

Diplomarbeit

Unterstützung von Verwaltungsaufgaben durch E-Governance am Beispiel eines web-basierten Abstimmungssystems

Michael Köster

Unterstützung von Verwaltungsaufgaben durch E-Governance am Beispiel eines web-basierten Abstimmungssystems

Diplomarbeit Informatik nach der Diplomprüfungsordnung 4, Universität Paderborn

Michael Köster, Matrikelnummer 6050759, michael.koester@uni.de

Betreut durch: Prof. Dr. Gerd Szwillus

Alle Quelltexte, Binärdateien, die Entwicklungsumgebung, eine Testumgebung und weitere Dokumentation, sowie eine digitale Version dieser Arbeit befinden sich auf der beigelegten CD.

Inhaltsverzeichnis

Teil I Vorwort

Der vorliegende Text ist das schriftliche Ergebnis meiner Diplomarbeit zum Thema *Unterstützung von Verwaltungsaufgaben durch E-Governance am Beispiel eines web-basierten Abstimmungssystems*.

Die Arbeit gliedert sich in zwei Teile, einen praktischen und einen theoretischen Teil. Der Praxisteil hat ein web-basiertes Abstimmungssystem als Ergebnis. Die Anwendung, seine Entwicklungsumgebung und eine Testumgebung befinden sich auf der beiliegenden CD. Eine Installationsanleitung und ein Einblick in die Benutzung der Testumgebung befinden sich am Ende dieser Arbeit, ebenso wie eine gedruckte Version des Online-Handbuches, welche die Bedienung der Webapplikation erklärt. Der Teil V dieser Arbeit – *Praxisbericht* – schildert die Entstehung des Produktes anhand der Chronologie eines Vorgehensmodells zur Softwareentwicklung, dem *Rational-Unified-Process.*

Die Theorie dieses aus der Praxis entliehenden Modells wird im Teil IV vorgestellt. Dieser Teil umfasst weiterhin einige theoretische Betrachtungen des Software-Engineerings im Webumfeld, die für die Entstehung der Anwendung relevant sind.

Der Hintergrund des Themas wird unter dem Stichwort *Abstimmungstheorie* im dritten Teil erarbeitet. Es wird eine Begriffsbestimmung vorgenommen. In unmittelbarem Zusammenhang mit dem Thema eines Abstimmungssystems zur Unterstützung von E-Governance steht die Betrachtung des Begriffes der *Elektronischen Demokratie*. Interessant ist an diesem Begriff seine Interdisziplinarität. Die technischen Herausforderungen gehen mit juristischen Fragestellungen genauso einher, wie mit gesellschaftspolitischen und soziologischen. Der Teil III beleuchtet einige dieser Aspekte aus der informatischen Sicht.

Teil II Einführung

Infolge der zunehmenden Technisierung, der Verbreitung von ständigen und kostengünstigen Internetzugängen und dem Einzug der neuen Technik auch in Rathäuser und Ministerien wuchs der Wunsch nach der Realisierung von Online-Wahlen, also der Stimmabgabe zu einer regulären Wahl über das Internet anstelle der Stimmabgabe in einem Wahllokal. Die grundlegenden technischen Probleme scheinen dabei moderner Signatur- und Verschlüsselungstechnologie lösbar[1]. Diese Ansicht hat sich auch in Wissenschaft und Politik durchgesetzt. Parteien fordern daher die Einführung einer Option zur Online-Wahl, mit dem Hinweis auf die Notwendigkeit die technischen Hürden nicht nur prinzipiell lösbar zu machen, sondern auch praktikabel umzusetzen[2]. Das Bundesministerium des Inneren hatte indes bereits die Umsetzung einer Online-Wahl für das Jahr 2006 angekündigt, dies allerdings dann auf 2010 verschoben. Entsprechende Projekte werden von der Bundesregierung gefördert. Das Projekt „Wahlen im Internet“ dabei bereits Anfang des Jahrtausends mit etwa TEUR 665 (1,3 Mio. Deutsche Mark) durch die Bundesregierung. Das damit verbundene Ziel, *die Ereignisse breitenwirksam, unter anderem durch Veranstaltungen und Dokumentationen, aufzubereiten*[3] zeigt aber das eigentliche Problem. Es ist nicht die technische Umsetzung, sondern die Schaffung von Akzeptanz der neuen Möglichkeiten. Der Eingriff in das zentralste Element einer indirekten repräsentativen Demokratie, der Vertreterwahl, muss wohlüberlegt stattfinden.

Unbestritten ist die Tatsache, dass mit digitalen Vernetzung, auch die Chancen für verbesserte politische Kommunikation steigen. Dies hielt bereits die Enquête-Kommission in ihrem Schlussbericht fest[4]. Der Austausch über das Netz zwischen Bürgern und staatlichen Stellen wie auch zwischen Bürgern direkt wurde verbessert.

1 Siehe Skrobotz, Das elektronische Verwaltungsverfahren, S. 38ff.

2 Skrobotz, Das elektronische Verwaltungsverfahren S. 36f mit Verweisen auf Anträge von Bündnis 90/Die Grünen, der FDP. der PDS, der SPD und dem parlamentarischen Staatssekretär des Bundesinnenministeriums, sowie einem Antrag der CDU/CSU Bundestagsfraktion.

3 Skrobotz, Das elektronische Verwaltungsverfahren, S. 35.

4 Erläuterung Enquête-Kommission, siehe Medien-Enquête, BT-Drs. 13/11004, S. 79f.

Abbildung 1: Schlagworte rund um den Begriff der E-Governance

1.1 Eingrenzung

Der Titelteil „Unterstützung von Verwaltungsaufgaben durch E-Governance“ gibt Anlass zur Klärung der Begrifflichkeiten. Was ist E-Governance? Verspätet, aber noch immer im Zuge des Hypes um die digitale Revolution, findet sich in der Literatur eine große Zahl von Neologismen rund um den Begriff der E-Governance. Faktisch lassen sich dabei nur schwer eindeutige und durchgängige Definitionen der vielfältigen Begriffslandschaft identifizieren. Am treffensten ist die Wortmenge durch eine Schlagwortwolke zu erfassen. Abbildung 1 zeigt eine solche Übersicht mit genannten Begriffen und einer subjektiven Gewichtung. Die Arbeit wird zunächst die Begriffe anhand der Literaturlage einordnen, soweit dies möglich ist.

Es wird sich herausstellen, dass die in der Literatur getroffenen Betrachtungen häufig mit dem Augenmerk einer auf einer realen Wahl und insbesondere einer Wahl von Personen in ein öffentliches Amt geschehen. Aus der vielfältigen und uneinheitlichen Darstellung der Begrifflichkeit gilt es also diejenigen Betrachtungen zu extrahieren, die exponierte Relevanz für eine auf den Vorgang einer Abstimmung haben. Dazu werden zunächst verschiedene Formen von Meinungsbildungen oder demokratischen Systemen dargestellt. Ein demokratisches System ist dabei jede Form von Meinungsbildung, die auf der gleichgewichteten Stimmabgabe einer definierten Personenmenge beruhen. Dazu gehören Wahlen, genauso wie Abstimmungen in Gremien. Eine vergleichbare Klassifizierung ist nach dem Kenntnisstand der in dieser Arbeit untersuchten

Literatur bisher nicht vorgenommen worden.

Davon zu unterscheiden ist das Verfahren, in der die Stimmen gesammelt und ausgewertet werden. Zu diesem Thema hat sich in der Literatur eine recht einheitliche Darstellung herausgebildet.

1.2 Softwaretheorie

Alle in dieser Arbeit untersuchte Literatur verharrt in theoretischen Überlegungen. Belege für die Machbarkeit der dargestellten Modelle werden nicht gegeben. Für existierende Beispiele für elektronische Wahlsysteme werden Schwächen aufgezeigt. Eine solche Schwäche wurde zuletzt im Zusammenhang mit einem Urteil des Bundesverfassungsgerichtes vom 3. März 2009. Dieses hatte die Nicht-Verfassungsgemäßheit von elektronischen Wahlgeräten festgestellt, welche bei der 16. Wahl zum Deutschen Bundestag eingesetzt worden waren. Ulrich Karpen , Professor für Rechtswissenschaften an der Universität Hamburg hatte diese Schwäche bereits 2005 in Buchform dargestellt[5].

Was sind also die Notwendigkeiten für ein verfassungsgemäßes Abstimmungssystem? Die Arbeit stellt einige Aspekte dieser juristischen Frage dar. Im Gegensatz zur juristischen Literatur möchte sich das Projekt dem Problemkomplex aber von einer anderen Seite nähern. Ziel ist zunächst eine Webanwendung zu realisieren, welche nach den Erkenntnissen des Software-Engineering mit einer sauberen Architektur geplant ist. Dies umfasst den Einsatz der richtigen Technik genauso wie die Gestaltung der Oberfläche. Dabei sind die Restiktionen des Einsatzes im Webumfeld zu beachten. Die theoretischen Überlegungen dazu werden dem konkreten Projekt vorangestellt.

5 Siehe Karpen, Elektronische Wahlen? - Einige verfassungsrechtliche Fragen, S. 31ff.

Teil III Abstimmungstheorie

Das Kapitel lichtet den Nebel um die Schlagwortwolke rund um den Begriff der E-Governance anhand der Literaturlage. Die Betrachtungen arbeiten sich dabei von diesem sehr abstrakten Begriff hinunter bis zum Begriff des I-Votings. Dieser Durchstich schafft den Rahmen für das Thema dieser Arbeit.

Darauf aufbauend erfolgt eine Klassifikation von Abstimmungsverfahren, insbesondere von Verfahren der elektronischen Demokratie einerseits und von Abstimmungsformen zur Abgrenzung der Abstimmungen von Wahlen zu öffentlichen Ämtern andererseits.

Zuletzt werden einige wichtige rechtliche Aspekte genannt und deren Auswirkungen auf die Gestaltung eines elektronischen Abstimmungssystens. Es wird beleuchtet, welche Chancen ein solches System auf den tatsächlichen Einsatz in demokratischen Willensbildungen hat.

1. Begriffsbestimmung

Der interessierende Themenkomplex der *Unterstützung von E-Governance durch ein web-basiertes Abstimmungssystem* wird durch eine große Menge von Schlagworten charaktisiert. Studien beschäftigen sich vielfältig mit dem Komplex des E-Government und der E-Administration, Begriffe wie das Virtuelle Rathaus und die E-Forms, also digital bearbeitete Verwaltungsanträge werden betrachtet. Gegenstand der rechtwissenschaftlichen Literatur ist elektronische Demokratie im Allgemeinen und E-Voting im Besonderen. Hier interessiert vor allem die Rechtmäßigkeit von elektronischen Wahlen. Deren Ein- und Durchführung ist ebenfalls Gegenstand von soziologischen Untersuchungen. Es wird die Veränderung des gesellschaftlichen Demokratieverständnisses durch die Verbesserung des Zuganges zur Entscheidungsfindung diskutiert. Die Disziplin Informatik untersucht einerseits die technische Umsetzbarkeit von Wahlautomaten, insbesondere im Hinblick auf Verschlüsselungstechnologien zu Sicherung der Übertragung und dem Einblick in Wahlentscheidungen (Geheimnisprinzip). Andererseits versuchen Publikationen der Rubrik "Informatik und Gesellschaft" Ordnung in die Nomenklatur zu bringen. Existierende Ordnungssysteme sind allerdings nicht komplett und widersprechen sich zudem.

Alle genannten Begriffe werden in der Literatur als aktueller Themenkomplex verstanden. Diese Einschatzung gilt fakultätsübergreifend in Rechtswissenschaft, Sozialwissenschaft und Informatik.

E-Goverment

Der Begriff des E-Goverment hebt sich unter den gegebenen Schlagworten hervor, denn er ist im Vergleich umfangreicher untersucht und war Gegenstand auch wissenschaftlicher Untersuchungen. Es existiert die sogenannte *Speyerer Definition von Electronic Government* der Autoren Reinermann und von Lucke[6]:

> Unter Electronic Government verstehen wir die Abwicklung geschäftlicher Prozesse im Zusammenhang mit Regieren und Verwalten (Government) mit Hilfe von Informations- und Kommunikationstechniken über elektronische Medien ... Diese Definition umfasst sowohl die lokale oder kommunale Ebene, die regionale oder Landesebene, die nationale oder Bundesebene sowie die supranationale und globale Ebene. Eingeschlossen ist somit der gesamte öffentliche Sektor, bestehend aus Legislative, Executive und Jurisdiktion sowie öffentlichen Unternehmen.

6 Herausgegeben vom Forschungsinstitut für öffentliche Verwaltung an der Deutschen Hochschule für Verwaltungswissenschaften in Speyer. Jörn von Lucke und Heinrich Reinermann beschäftigen sich darüber hinaus in einigen Publikationen und Assays mit dem Thema E-Government. Siehe dazu beispielsweise Büchner und Büllesbach, E-Government.

Autoren angrenzender Themen übernehmen diese Definition[7]. Andere Quellen zitieren die Definition, aber fassen den Begriff explizit weiter als „jeglichen staatlichen Handelns durch elektronische Medien“[8]. Diese Formulierung erweitert die rein transaktionale Speyerer Definition von E-Government. Diese erweiterte Sichtweise wird auch als *E-Government im weiteren Sinne* beschrieben[9]. Im Gegensatz dazu steht *E-Government im engeren Sinne*, welche lediglich die Nutzung moderner Medien wie das Internet durch die öffentliche Verwaltung meint. Dieser Aspekt wird auch mit dem Begriff E-Administration belegt[10].

E-Government (im weiteren Sinne) ist damit ein Überbegriff. Dieser wird mehrdimensional untergliedert, zum einen in der Interaktionsstufe. Hier wird zwischen der Bereitstellung von unidirektionaler Information (E-Information[11]) für die Bevölkerung, der Möglichkeit zur digitalen Kommunikation (E-Communication) per E-Mail, Chat oder durch webbasierte Foren und drittens in einer transaktionalen Ebene (E-Transactions) unterschieden. Hier beinhaltet Electronic-Government Formularlösungen (E-Forms), welche nach Druck und Unterschrift einzusenden sind oder intelligente Online-Lösungen mit unmittelbarer Eingabeverifikation und – automatisierter - Rückmeldung an den Benutzer.

Einige wirtschaftlich orientierte Publikationen untergliedern E-Government anhand der handelnden Personen und Institute[12]. In Anlehnung an Begriffe aus dem E-Business werden vier handelnde Sektoren unterschieden.

1. Der Bürger/die Bevölkerung (im Modell *Citizen*)
2. Der Staat/die Verwaltung (im Modell *Government*)
3. Die Privatwirtschaft, der „Zweite Sektor“ (im Modell *Business*)
4. Sogenannte Non-Profit-Organisationen/Non-Government-Organisationen, welche in Deutschland volkswirtschaftlich als „Dritter Sektor“ bekannt sind (im Modell *NPO/NGO*)

---

7 Siehe Schönau, Elektronische Demokratie

8 Siehe Skrobotz, Das elektronische Verwaltungsverfahren, S. 22f.

9 So auch bei Gantert, E-Demokratie.

10 Siehe Skrobotz, S. 22 mit Verweis auf Alexander Roßnagel und explizit bei Gantert, E-Demokratie.

11 Trotz ihres Schlagwortcharakters werden die Begriffe E-Information, E-Communication, E-Transaction, E-Forms und weitere wenig kantenreiche Neologismen von der Speyerer Definition explizit genannt.

12 Zu finden in E-Government und der moderne Staat, herausgegeben von Booz Allen Hamilton.

	Citizen	Government	Business	NPO/NGO
Citizen	C2C	C2G	C2B	C2N
Government	G2C	G2G	G2B	G2N
Business	B2C	B2G	B2B	B2N
NPO/NGO	N2C	N2G	N2B	N2N

Die Gliederung zeigt den umfassenden Ansatz von E-Government, alle staatlichen Beziehungsgeflechte und damit sieben der sechszehn Beziehungen werden abdeckt[13].

Organisatorisch werden dem Überbegriff E-Government die Teilmengen E-Administration und E-Democracy zugeordnet. Diese Teilmengen sind jedoch nicht disjunkt, Literatur über Elektronische Demokratie erwähnt durchgängig E-Information als einen wichtigen Aspekt[14], welcher ebenso Teil der E-Administration ist. Soll E-Government neben Executive und Legislative auch Jurisdiktion umfassen, so ist die Unterteilung nicht vollständig. Es ist nicht ersichtlich, inwieweit E-Administration als Aufgabe der öffentlichen Verwaltung E-Justice als Teilmenge von E-Government erfassen kann. Die Speyerer Definition schließt E-Justice implizit mit ein. Eine Erläuterung, wie EGovernment oder E-Administration in der rechtssprechenden Gewalt Anwendung finden kann, fehlt.

Nicht immer wird Elektronic Government als Überbegriff bewertet. Literatur über elektronische Demokratie weist insbesondere auf den durchgängigen Charakter von demokratischen Entscheidungsprozessen in unserer Gesellschaft hin. Im Zuge der Digitalisierung von Information, Diskussion, Partizipation ist nach diesem Ansatz auch die elektronische Organisation der Verwaltung der Demokratie (E-Government) eine Teilmenge der Elektronischen Demokratie (E-Democracy)[15]. In dieser Hinsicht kehrt sich die Anhängigkeit der beiden Disziplinen gerade um.

1.1 E-Democracy

E-Democracy wird im Gegensatz zu anderen Schlagworten in der deutschen Literatur häufig in der Übersetzung „Elektronische Demokratie“ beschrieben. Einige Quellen betrachten E-Democracy als die „Input“-Seite von E-Government. Dem stünde in diesem Bild E-Administration die elektronische

13 Details zu G2C, G2B, G2G und Umkehrbetrachtungen siehe Booz Allen Hamilton. G2N und N2G nur in der Speyerer Definition dargestellt, ohne Beispiele. Martin Schmalbruch in E-Government [P11] unterscheidet außerdem mit Government-to-Employee (G2E) bezüglich der Beziehung einer Behörde zu den eigenen Mitarbeitern.

14 Siehe beispielsweise Holznagel/Grünwald/Hanßmann, Elektronische Demokratie, Schaubild S. 18, aber auch Schönau, Elektronische Demokratie und weitere.

15 Siehe dazu ausführlich Schönau, Elektronische Demokratie, S. 19ff und S. 31f.

„Output“-Komponente[16] gegenüber, welche dem Bürger öffentliche Leistungserbringung offeriert.

Besonders rechtswissenschaftliche Literatur wird ebenfalls auf den umfassenden Charakter von Demokratie im Allgemeinen und E-Democracy im Besonderen hingewiesen[17]. Demnach sind Elemente einer elektronischen Demokratie E-Information, E-Discussion, E-Partizipation, E-Protest, E-Party, E-Activision und weitere. Nicht alle Autoren verwenden dabei „E-Begriffe“ explizit, beschreiben aber im Kern die gleichen Inhalte. Unisono wird unter E-Democracy die Abbildung des demokratischen Prozesses in die digitale Welt verstanden.

Definitionen des Begriffes lassen sich bei Gantert [P06] und in Onlinequellen finden[18]. Nach Gantert wird unter elektronischer Demokratie der „Einsatz des Internets im politischen Meinungs- und Willensbildungsprozess sowie die Realisierung einer aktiveren Bürgerbeteiligung, auch im Bereich von Wahlen“ verstanden. Und mit Verweis auf die Internetseite des Bundesinnenministeriums hinzugefügt, dass „Elektronische Demokratie sämtliche Maßnahmen zusammenfasst, 'bei denen Internettechnologien eingesetzt werden, um Bürgerinnen und Bürgern zusätzliche demokratische Mitbestimmungs- und Gestaltungsmöglichkeiten einzuräumen.'“.

Ein wesentliches Element von E-Democracy ist zunächst die politische Bildung, also Information. E-Information ist damit schneller als bisher zugängliche Bereitstellung von Informationen im Netz. Einige Quellen heben eine Sonderform von E-Information besonders heraus, die Webseiten von politischen Parteien und Abgeordneten. Dies wird gelegentlich als E-Party bezeichnet.

Logisch anschließend folgt: Der Diskurs über gegebene Informationen als zweite tragende Säule im demokratischen Willensbildungsprozess. E-Discussion fördert Auseinandersetzung in drei Dimensionen.

1. Im direkten Austausch zwischen Bürgern zum Beispiel in Diskussionsforen oder Kommentarfeldern zu politischen Artikeln.

2. Im Austausch zwischen Bürgern und politischen Veranstwortungsträgern. Hier verrinert digitale Technik den gefühlten Abstand durch Online-Chats nach Ende von politischen Talkrunden oder Kontaktformulare auf Webseiten von Abgeordneten.

3. Vermehrt auch in der Kommunikation im **Internetwahlkampf**. CDU und SPD hatten in den Bundestagswahlkämpfen 2002 und 2005 jeweils Webseiten geschaltet, die unmittelbar auf Wahlkampfaussagen des jeweiligen politischen Gegners antworteten[19].

Der nächste Schritt ist E-Participation, also die unmittelbare Beteiligung und

16 So in E-Government und der moderne Staat, Booz Allen Hamilton, Kapitel 8, „E-Government als elektronische Demokratie“
17 Besonders umfangreich bei Schönau, Elektronische Demokratie
18 Siehe http://www.egov-conference.org/glossary
19 Für die CDU war es www.wahlfakten.de und für die SPD die Seite www.die-falsche-wahl.de. Beide Seiten sind Stand heute (Februar 2009) inaktiv.

Einflußnahme. E-Participation bezeichnet dabei ausdrücklich nicht den eigentlichen Beteiligungsprozess, den Wahlvorgang, sondern Beteiligung **während** einer Legislaturperiode[20]. Es werden hierfür Bürgerforen und Online-Umfragen genannt. In Deutschland schließt sich daran eine Diskussion über Volksabstimmungen im Allgemeinen und die Auswirkungen durch Hinzunahme von digitaler Technik auf den Entscheidungsprozess im Besonderen an[21].

Davon zu unterscheiden sind Aktionen unter dem Stichwort EProtest. Diese dienen nicht der direkten politischen Kommunikation oder Partizipation, sondern einer indirekten Einflussnahme durch die Erregung von breiter Aufmerksamkeit. Diese Form von E-Democracy läßt sich am ehesten mit der Durchführung von Demonstrationen und Kundgebungen vergleichen, erreicht aber erst durch die Möglichkeiten des Internets mit vordigitalen Zeiten Verbreitung und mitunter starkes Medieninteresse[22]. Nicht scharf ist die Abgrenzung von E-Activism zu E-Protest[23], die Begriffe werden häufig synonym verwendet..

Zum Schlagwort E-Parliament liefert der *World E-Parliament Report 2008*[24] eine Definition. Demnach dient E-Parliament dem „verbesserten Zugang zu parlamentarischen Dokumenten und Aktivitäten“. Es wird explizit der Vorteil sowohl für Wähler, als auch für Gewählte genannt.

Der World E-Parliament Report 2008 liefert eine Bestandsaufnahme über den weltweiten Einsatz von IT in parlamentarischer Arbeit. Im Kern wird darin länderübergreifend ein großes Potential erkannt und teilweise auch schon heute genutzt[25]. Genannt werden Dokumentenmanagementsysteme, Bemühungen um einheitliche, offene Datenstandards[26], das Angebot inhaltsreicher Webseiten, die Nutzung verschiedener Kommunikationskanäle wie Video und Potcast neben Text und insgesamt der Versuch die Onlinequellen nah an aktueller Gesetzgebung zu halten. Der Report deckt aber auch die signifikante Lücke zwischen Potential der Technologie und tatsächlichem Einsatz auf. Insbesondere die Idee eines elektronischen Abstimmungssystem zur Arbeitsoptimierung wird allerdings nicht erwähnt.

---

20 Siehe Schönau, Elektronische Demokratie.

21 Siehe dazu insbesondere „E-Demokratie = Ende der Demokratie?“ aus der Schriftenreihe Forum Medienethik, herausgegeben von der Diözese Rottenburg-Stuttgart.

22 Ein Beispiel ist die Begleitung der Einführung von Studiengebühren/Studienbeiträgen. Diese wurde sowohl von umfangreichen Protestseiten des „Aktionsbündnisses gegen Studiengebühren“ [U02] als auch von Befürwortern [U03] gestaltet.

23 Überhaupt eine getrennte Nennung beider Begriffe ließ sich einzig innerhalb des Artikels E-Democracy [W06] der Wikipedia finden.

24 Der World E-Parliament Report 2008 wurde im Zuge der World E-Parliament Conference erstellt. Die Konferenz wird jährlich vom Global Centre for ICT in Parliament [U04] organisiert, einer Unterorganisation der Vereinten Nationen. Die Abkürzung ICT steht dabei für *Information and Communication Technologies*.

25 Siehe World E-Parliament Conference 2008, Background Document, S. 2f, zu finden auf der Internetpräsenz des Global Centre for ICT in Parliament [U05].

26 Untersuchungsgegenstand der Konferenz 2008 ist der Einsatz von Textdokumenten im XML-Standard zum verbesserten Datenaustausch. Genannt werden aber auch weitreichendere Beispiele, wie der Einsatz von freien Betriebssystemen (Linux) anstelle des vorherschenden Micosoft Windows.

E-Parliament als immanenter Bestandteil der E-Democracy enthält damit Elemente von E-Information und E-Communication genauso wie einen Ansatz von E-Administration für die Verwaltung der Parlamente. E-Parliament bezieht sich damit aber weniger auf die Organisation von Parlamentarismus im Allgemeinen, denn auf nationale und supranationale Parlamente. Zeitliche Einschränkungen oder Kostendruck sind auf dieser Ebene kein bestimmender Faktor und daher haben elektronische Abstimmungssysteme keine Notwendigkeit.

Der Begriff des E-Voting wird in der deutschsprachigen Literatur weitgehend als elektrische Wahl übersetzt, obwohl die Vokabel „vote“ (Abstimmung, Wahl) dies nicht unbedingt fordert. Untersuchungsgegenstand ist also die Automatisierung von parlamentarischen Wahlen[27]. Aber auch „andere Wahlen“ werden explizit genannt, etwa die Personalratswahlen, Wahlen zu Hochschulgremien und Wahlen im Bereich der Wirtschaft, wie Aktionärswahlen[28]. Teilweise gelten die Überlegungen in Bezug auf die Wahl analog auch für Abstimmungen[29], was zum Beispiel die Einhaltung von geheimer Wahl oder Abstimmung angeht[30]. E-Voting umfasst damit jede Form von Stimmabgabe auf elektronischem Wege, also Wahlen, Abstimmungen, Umfragen, Befragungen oder Erhebungen.

Die elektronische Wahl kann grundsätzlich in verschiedenen Formen auftreten. Neben einer Onlinewahl über das Internet vom heimischen Computer sind sogenannte Wahlkioske denkbar, die das elektronische Wahlequipment bereitstellen oder auch nur die elektronische Auswertung von händisch ausgefüllten Wahlzetteln[31]. Einige Autoren verwenden den Begriff E-Voting nur für die sogenannte Internetwahl. E-Voting ist dann gleichbedeutend mit I-Voting, Online-Voting und Cyber-Voting[32].

27 Zum Beispiel bei Robert Schönau, Elektronische Wahlen.
28 Siehe dazu Ulrich Karpen, Elektronische Wahlen? - Einige verfassungsrechtliche Fragen, S. 42ff.
29 Zur Übertragbar der Überlegungen unter dem Titel Internetwahl auf Internetabstimmungen siehe Martin Will, Internetwahlen, S. 15f.
30 Gantert, Elektronische Demokratie. Es wird eine gesonderte Betrachtung von Abstimmungen als zweite Form der vom Volk ausgeübten Staatsgewalt nach Art. 20 Abs. 2 des Grundgesetzes vorgenommen. Gegenstand der Betrachtungen sind also Referenden und Bürgerentscheide.
31 Für eine detaillierte Untergliederung der Stufen von elektronischen Wahlen siehe Abschnitt III.2 Abstimmungsverfahren.
32 So zum Beispiel bei Ulrich Karpen, Elektronische Wahlen, S. 11.

2. Abstimmungsverfahren

Bestrebungen den Vorgang der Meinungsbildung zu automatisieren existieren schon lange. Es lassen sich dabei verschiedene Grade der Automatisierung unterscheiden. Die rechtliche, soziologische und technische Beurteilung der Rahmenbedingungen solcher Systeme unterscheiden sich. Daher gibt dieses Kapitel eine Übersicht der Klassifikationen. Die Darstellung folgt dabei Schönau [P14] und der sehr ausführlichen Darstellung von Khorrami [P20].

2.1 Mechanische Systeme

Zunächst existierten Systeme, welche rein mechanisch arbeiten. Denkbar ist ein manuell ausgelöstes Zählwerk in der Form eines „Wahlhebels“. Solches Systeme wurden in den USA bereits 1892 entwickelt und sind unter dem Namen *Mechanical Lever Machines* bekannt. Im Fall von Abstimmungen gibt es analog improvisierte Systeme, welche durch den Einwurf einer Wahlmünze in eine Urne funktionieren. Die Wägung der Urne bringt das Ergebnis. Solcherlei Ansätze haben sich in Deutschland nicht durchgesetzt, weder als ernsthafte Wahlhilfe, noch in der improvisierten Variante. Die Anschaffungs- und Lagerungskosten dieser Geräte stehen nur einem geringen Geschwindigkeitsgewinn und einem geringen Reduzierungspotential in der Anzahl freiwilliger Helfer bei einer Wahl gegenüber[33].

2.2 Elektronische Stimmenzählgeräte

Der nächste der Schritt der Automatisierung ist der Einsatz elektronischer Stimmenzählgeräte[34]. Dabei werden Stimmzettel ausgefüllt oder speziell markiert und anschließend machinell gelesen und elektronisch ausgewertet. In den USA existieren solche System schon seit den sechziger Jahren. Beim *Punchcard*-System beispielsweise müssen computerlesbare Karten mit geeignetem Werkzeug gezielt gestanzt werden. Modernere Systeme benutzen eine optische Scanmethode. Optische Systeme lesen einen herkömmlichen ausgefüllten Stimmzettel zum Ankreuzen.

2.3 Wahlcomputer

In gewisser Hinwicht die Vereinigung der beiden Ideen der *Mechanical-Lever-Machines* und der optischen Scanner stellen Wahlcomputer dar. Solche Systeme werden auch als *Direct Electronic Recording (DER)* bezeichnet. Es wird kein Stimmzettel verwendet, die Stimmabgabe erfolgt über die Bedienoberfläche der Geräte. Diese kann entweder in Form von Knöpfen ausgeführt sein, oder durch eine digitale Oberfläche auf einem Touchscreen. Die Hardware ist entweder ein handelsüblicher PC mit spezieller Software oder eine PC-unabhängige

33 Siehe Schönau, Elektronische Demokratie, S. 51.
34 So genannt bei Schönau. Khorrami nennt den englischen Begriff Machine Counting.

Konstruktion[35]. Die Geräte übernehmen sowohl die Funktion des Stimmzettels als auch der Stimmenauszählung.

2.4 Onlinewahl im Wahllokal

Eine weitere Automatisierung betrifft die Organisation des Wahlregisters, welches in den bisherigen Betrachtungen von der Automatisierung ausgenommen war. Die Stimmabgabe erfolgt wie bei einem Wahlcomputer an einem Computerterminal. Dieses Terminal ist jedoch mit einem zentralen Wahlregister vernetzt. Der Wähler kann daher in jedem beliebigen Wahllokal seine Stimme abgeben, ein entscheidender Vorteil gegenüber den zuvor genannten Verfahren. Die Onlinewahl im Wahllokal wird auch als *Polling Place Internet-Voting* oder *Poll-Site-Voting* bezeichnet. Die Onlinewahl im Wahllokal ist der erste Schritt zu einer vernetzten Wahl und geht damit in Richtung einer Distanzwahl. Dennoch verbleiben die eingesetzten Geräte vollständig unter der Kontrolle der eingesetzten Wahlvorstände.

2.5 Kiosk-Onlinewahl

Eine Weiterentwicklung der Onlinewahl im Wahllokal ist die Kiosk-Onlinewahl. Sie stellt einen Zwischenschritt zur reinen privaten Internetwahl dar. Als Wahlterminals kommen nicht mehr speziell in Wahllokalen aufgestellte Geräte zum Einsatz, sondern öffentlich zugängliche und vernetzte Computer. Der Vorteil gegenüber den Onlinewahlen im Wahllokal ist die Möglichkeit auf bestehende Infrastruktur zurückzugreifen, wie sie beispielsweise in Bibliotheken oder öffentlichen Verwaltungen gegeben ist. Dem steht ein gewisser Kontrollverlust der Wahlvorstände über die eingesetzten Systeme gegenüber und die größere Angreifbarkeit der Infrastruktur (aufgrund der klassischen PC-Architektur und der Angriffsmöglichkeiten auch außerdem der Wahltermine).

2.6 Private Internetwahl

Die private Internetwahl stellt nach heutigen Stand das Maximum an Automatisierung einer Abstimmung dar. Es ist kein Besuch einer Wahlstelle notwendig. Die Stimmabgabe ist über jedem Internetzugang möglich und schränkt sich damit nicht einmal mehr auf PC-Architektur ein, genauso gut wären (wenigstens prinzpiell) Handys, Netbooks, Automobile, etc. denkbar. Die verwendeten Geräte stünden unter keinerlei Kontrolle der Wahlvorstände, und die Wahl wäre eine reine Distanzwahl im räumlichen Sinne. Bei der privaten Internetwahl muss die Wahlstelle aber nicht zwingend die eigene Wohnung sein. In Frage kommen natürlich auch Internetcafés, der Arbeitsplatz oder öffentlich zum Zwecke der Wahl bereitgestellte Internetzugänge, um Stimmberechtigten ohne Internetzugang den Wahlgang zu ermöglichen. Die private Internetwahl wird auch *Remote-Internet-Voting* oder *Private-Remote-Internet-Voting* genannt.

---

35 Die in einigen Gemeinden Deutschlands eingesetzten Wahlcomputer und vom Bundesverfassungsgericht als als unvereinbar mit dem Grundgesetz beurteilten Geräte sind solche PC-unabhängigen Wahlcomputer, die mit Knöpfen bedient werden. Siehe Schönau, Elektronische Wahlen, S. 52ff. und Kapitel III.5 Rechtliche Aspekte.

3. Abstimmungsformen

Eine Abstimmung im eigentlichen Wortlaut ist das Abgeben seiner Stimme oder Meinung zum Zwecke einer Mehrheitsbildung. Demnach ist die Entscheidung über einen Antrag genauso eine Abstimmung wie die klassische Wahl zu einem öffentlichen Amt eine Abstimmung ist. Dieser Abschnitt dient der Begriffbildung. Quelle der folgenden Definitionen ist eine Gedankenspielerei – ein Brainstorming – sie entspringen nicht der Literatur. Dies hat seine Ursache darin, dass wissenschaftliche Literatur zum Beispiel im Bereich der Politikwissenschaften Wahlen und Abstimmungen und deren Begleitumstände aufgrund der tiefgreifenden Bedeutung für die demokratische Gesellschaft wesentlich ausführlicher analysieren, als dies für die Schaffung der Nomenklatur für ein webbasiertes Abstimmungssystem notwendig wäre.

Weiterhin sind die Abgrenzungen zwischen einzelnen Begriffen im deutschen Sprachgebrauch nicht eindeutig. Unter einer Abstimmung wird vornehmlich die Stellungnahme zu einem Antrag verstanden. Dem Stimmberechtigten stehen dabei die Optionen *Ja*, *Nein* und *Enthaltung* zur Verfügung. Auch eine Wahl kann dann als Sonderform der Abstimmung gestaltet sein, nämlich im Falle nur eines Kandidierenden. Konfusion auch bei der Übertragung in den englischsprachigen Raum. Nach Wikipedia [W12] beispielsweise entspricht der deutsche Begriff der *Abstimmung* dem englischen Begriff *Vote.* Dieser umschreibt dem Text nach jedoch eher einen Überbegriff über Abstimmungen **und** Wahlen.

3.1 Definition Wahl

Eine **Wahl** ist die Entscheidung über die Auswahl einer natürlichen Person oder mehrerer natürlichen Personen für ein Amt oder ein Gremium. Ist mehr als eine Position zu besetzen, so handelt es sich um eine Blockwahl. Eine Blockwahl hat immer mehr oder genauso viele Kandidaten wie Positionen zu besetzen sind. Im Falle von mehreren Kandidaten hat jeder Stimmberechtigte genau eine Stimme, die nicht abgegeben werden muss. Dies kommt einer Enthaltung gleich. Im Falle von einem Kandidaten für ein Amt haben Stimmberechtigte die Option der Zustimmung, Ablehnung und Enthaltung.

3.2 Definition Abstimmung

Eine **Abstimmung** ist die Stellungnahme von Stimmberechtigten über eine gegebene Fragestellung – ein Antrag oder ein Vorschlag. Die Stimmberechtigten haben dabei die Optionen der Zustimmung, Ablehnung und Enthaltung. Jeder Stimmberechtigte hat genau eine Stimme. Er kann auf die Abgabe verzichten. Dies kommt einer Enthaltung gleich.

3.3 Definition Umfrage

Eine **Umfrage** ist die Stellungnahme von Stimmberechtigten über eine gegebene

Fragestellung. Die Fragestellung bezieht sich auf die Meinung des Stimmberechtigten. Der Stimmberechtigte gibt eine gewichtete Meinung zur Fragestellung ab. Die Skala für die Gewichtung ist ein Intervall der ganzen Zahlen, zum Beispiel das Intervall von Eins bis Sechs (Schulnotensystem). Ein anderes Beispiel ist das Intervall von minus drei bis drei zur Abdeckung eines Spektrums von völliger Ablehnung über neutrale Meinung bis hin zur völligen Zustimmung. Jeder Stimmberechtigte hat genau eine Stimme. Nicht abgegebene Stimmen werden nicht gewertet.

3.4 Definition Poll

Bei einem **Poll** wählen Stimmberechtigte aus einer Menge von Alternativen ihren Favoriten aus. Die Alternativen bestehen dabei Bezeichnern oder ganzen Sätzen. Ein Beispiel ist die Frage: „Wer wird Deutscher Meister?“ mit den Alternativen deutscher Fußballbundesligaclubs. Der von der Fernsehsendung *„Wetten-Dass...?“* durchgeführte *TED*[36] ist in diesem Sinne ein Beispiel für einen Poll. Im Unterschied zum TED hat ein Stimmberechtigter bei einem Poll nur eine Stimme. Nicht abgegebene Stimmen werden nicht gewertet.

3.5 Überbegriff

Da unter dem Begriff Abstimmung damit bereits die Stellungnahme zu einem Antrag mit Ja, Nein, Enthaltung gemeint ist, kommt er als Überbegriff nicht in Frage. Zusammengefaßt wird daher unter einer **Mehrheitsmeinungsbildung** jede Form von Verfahren verstanden, bei denen die Stimmen Einzelner zur Bildung einer generischen Meinung Vieler kommuliert werden. Mehrheitsmeinungsbildung umfasst damit Wahlen, Abstimmungen, Umfragen und Polls. Mehrheitsmeinungsbildung kann kurz als **Meinungsbildung** bezeichnet werden. Systeme, welche den Prozeß der Meinungsbildung unterstützen, heißen **Demokratische Systeme**.

3.6 Abgrenzungen

Die vorgenommenen Definitionen stellen eine Vereinfachung dar. Insbesondere Wahlen werden in der Literatur vielfältig klassifiziert. Insbesondere der Parteienbegriff wurde hier außer Acht gelassen. Daran schließen sich Begriffe wie Verhältniswahl im Gegensatz zur Mehrheitswahl und Gestaltungsmöglichkeiten von Wahllisten an. Wahlsysteme sorgen für die Umsetzung der reinen Stimmenzuordnung in konkrete Abgeordnetenzahlen. Die Allgemeinheit von Wahlen (kompetitive Wahlen) und die Gleichheit von Stimmen (Zensuswahl) bei einer Wahl führen zu weiteren Klassifizierungen[37].

---

36 Der Begriff TED steht für Tele-Dialog und bezeichnet ein vom ZDF in Zusammenarbeit mit der Deutschen Bundespost 1979 entwickeltes Tele-Voting-Verfahren. Der TED selbst kommt heute nicht mehr zum Einsatz, es existieren verbesserte digitale Verfahren. Der Begriff TED ist aber in den allgemeinen deutschen Sprachgebrauch übergegangen. Siehe dazu Wikipedia [W13].

37 Zu diesem Thema existiert vielfältige Litertatur in den Politikwissenschaften und der Geschichte. Ein Standardwerk ist Dieter Nohlen, Wahlrecht und Parteiensystem – Zur Theorie der Wahlsysteme, Opladen 1986.

Umfragen werden in der Forschung viel Allgemeiner als Befragungen von Zielgruppen verstanden. Dies umfasst das komplette Gebiet der Demoskopie genauso wie die gesamte empirische Sozialforschung mit eigenen Begriffbildungen[38].

Der Poll existiert im deutschen Sprachgebrauch kaum, nur Wikipedia kennt den Web-Poll. Im Zusammenhang mit „Videotextumfragen“ erscheint der Begriff außerdem. Der im deutschen Sprachgebrauch eingebürgerte Begriff des TED wird natürlich in erster Linie mit dem Kommunikationsmedium Telefon in Verbindung gebracht.TED ist auch ein eingetragenes Markenzeichen der Deutschen Telekom. Web-Polls, Videotextumfragen und dem TED ist gemein, dass die Gleichheit der Stimmen nicht überprüft wird oder überprüft werden kann. Es ist niemand daran gehindert mehrfach anzurufen und seine Stimme in diesem Sinne mehrfach abzugeben – im Gegenteil, dies scheint gerade bei Tele-Votings oder Anrufgewinnspielen im Fernsehprogramm ein gewünschter Effekt zu sein, da jeder Anruf für die Finanzierung des ausgeschütteten Gewinns sorgt.

Demokratie als Volksherrschaft impliziert den Begriff der Demos. Demos wird dabei als Begriff aus der soziologischen Forschung abgeleitet. Demnach ist Demos als Volksbegriff im Sinne eines politischen und rechtlichen Begriffes von Volk verstanden; mit Rückbezug zur attischen Bürger-Demokratie. Volk ist demnach ein soziales und politisches Gebilde[39]. Entsprechend ist ein *Demokratisches System*, ein System, welches einem sozialen Gebilde zur Herrschaft verhilft, also zur gemeinsamen Meinungsbildung und -Äußerung.

3.7 Zusammenfassung

Mit diesem Kapitel ist eine einheitliche Nomenklatur für die Form einer Mehrheitsmeinungsbildung geschaffen. Diese Formen sind nur Teilmengen der real existierenden Systeme. Sie können aber nun leichter in einem webbasierten System abgebildet werden.

38 Zu beiden Themen siehe Noelle-Neumann/Petersen, Alle nicht jeder – Einführung in die Methoden der Demoskopie, 3. Auflage, Springer, Berlin 2000.

39 Diese Definition von Demos ist die vorherrschende Interpretation von Demos in der Soziologie und geht zurück auf den Soziologen Emerich K. Francis. Siehe Francis, Ethnos und Demos, Duncker und Humblot, Berlin 1995.

4. Rechtliche Aspekte

Der Einsatz elektronischer Hilfsmittel bei der Ermittelung von Stimmergebnissen muss vor dem Hintergrund der geltenden Rechtslage analysiert werden. Eine besondere Bedeutung kommt dieser Betrachtung aufgrund der weitgehenden Auswirkungen rechtswidriger Wahlentscheidungen zu. Die Überprüfung der Rechtmäßigkeit elektronischer Wahlen muss stattfinden vor dem Hintergrund des Abstimmungsverfahrens[40] einerseits und der Abstimmungsform[41] andererseits. Im Einzelfall kann drittens auch die Abstimmungsumgebung[42] eine Auswirkung auf die rechtliche Beurteilung haben.

4.1 Wahlprinzipien

Einige grundlegende Wahlrechtsgrundsätze haben sich als Allgemeingut freiheitlicher Demokratien etabliert und sind beispielsweise im deutschen Grundgesetz festgeschrieben, aber außerdem in fast allen demokratisch-rechtsstaatlichen Verfassungen[43] oder auch in der Allgemeinen Erklärung der Menschenrechte[44]. Die folgende Darstellung hat einen Fokus auf das deutsche Wahlrecht. Danach gelten die Wahlrechtsgrundsätze zunächst für die Wahlen zum Deutschen Bundestag, den Länderkammern und den Räten in Kreisen und Gemeinden. Darüber hinaus gelten sie aber auch als allgemeines Rechtsprinzip für alle Wahlen im öffentlichen Raum[45]. Satzungen und Geschäftsordnungen dieser Parlamente regeln Abstimmungsverfahren und beziehen sich auf diese Wahlrechtsgrundsätze.

Allgemeinheit

Allgemeinheit einer Abstimmung verbürgt die Gleichheit im Zugang zur Abstimmung. Sie ist damit eng verwandt mit dem Grundsatz der Gleichheit. Im Bezug auf eine Wahl untersagt der Grundsatz der Allgemeinheit „den unberechtigten Ausschluss von Staatsbürgern von der Teilnahme an der Wahl überhaupt“[46]. Die Allgemeinheit einer Abstimmung ist von elektronischen Hilfsmitteln im Wahllokal nicht eingeschränkt oder nur insofern eingeschränkt,

40 Elektronische Zählmaschinen werden anders beurteilt als eine Abstimmung im Internet. Siehe Abschnitt III.2.

41 Eine Wahl hat ein von einer Gremienabstimmung verschiedenes Gewicht. Siehe Abschnitt III.3.

42 Es ist prinzipiell denkbar, dass Wahlrechtsgrundsätze bei einer Personalratswahl weniger streng gefordert werden, als bei einer Wahl zum Deutschen Bundestag. Der rechtsverbindliche Wahlordnung kann beispielsweise gezielt von grundgesätzlichen Regelungen abweichen, gerade um elektronische Wahlen zu erlauben.

43 Siehe Dieter Nohlen, Wahlrecht und Parteiensystem, Opladen 1990.

44 Siehe beispielsweise Bundeszentrale für Politische Bildung, Serie Pocket Politik, Demokratie in Deutschland, Bonn 2006.

45 BVerfGE 30, 227 (246).

46 BVerfGE 36, 139 (141); 58, 202 (205).

dass eine gewisse Hemmschwelle in der Benutzung der Systeme bestehen könnte[47]. Im Fall der Realisierung von elektronischen Distanzwahlen ergibt sich ein anderes Bild. Zunächst ist der Einsatz einer solchen Wahl in Ablösung der Präsenzwahl an der Urne mit dem Grundsatz der Allgemeinheit nicht vereinbar, da nicht jedem Wahlberechtigten ein entsprechender Zugang bereit steht. Selbst im Falle der Bereitstellung durch die öffentliche Hand, bestünden Bedenken im Hinblick auf die Zugangsmöglichkeiten technisch nicht versierter Stimmberechtigter. Auch im Falle einer ergänzenden Internetwahl stehen Zweifel in Bezug auf die Allgemeinheit insofern im Raum, als das Internetnutzer mit dem bequemeren Weg gegenüber Nicht-Internetnutzern im Vorteil stehen könnten. Eine Internetwahl als optionale Alternative zur Briefwahl ist jedoch unter dem Gesichtspunkt der Allgemeinheit zulässig[48].

Es bestehen weitere Bedenken im Hinblick auf die Übertragungs- und Verarbeitungssicherung per Internet abgegebener Stimmen. Für diese bestünde ein erheblich erhöhtes Sicherheitsrisiko nicht gewertet zu werden. Dies bedeutete eine Einschränkung der Allgemeinheit[49] der Wahl.

Konsequenzen

In Bezug auf den allgemeinen Einsatz von Internetabstimmungssystemen zur Wahl, Umfrage oder Abstimmung sollte daher im Sinne des Grundsatzes der Allgemeinheit davon ausgegangen werden, dass die jeweilige Abstimmung ausschließlich auf diesem Weg durchzuführen ist und keine alternativen Möglichkeiten der Präsenz- oder Briefwahl bestehen. Sollte eine Stimmabgabe durch technische Probleme auf der Seite des Abstimmungssystemanbieters nicht möglich sein, wäre ein Alarmsystem auf einem alternativen Kommunikationskanal wünschenswert (zum Beispiel E-Mail, besser Telefon).

Die Architektur des Abstimmungssystems sollte möglichst einfach sein, um die Zugänglichkeit zu gewährleisten. Dies umfasst die Verwendung einer nicht zu hohen Bildschirmauflösung, eines nicht zu feingranularen Farbschemas für die Anzeige auf Bildschirmen mit geringer Farbtiefe, den Verzicht auf Browser-spezifische Codierungen und keine Abhängigkeit von Zusatztechnologien wie *JavaScript* oder *ActiveX.*

Der Verlust der Allgemeinheit einer Abstimmung ist jedoch immanent durch die Technisierung des Vorganges gegeben. Eine Lösung ist die Bereitstellung von Zugangssystemen für alle Stimmberechtigten. Dies ist beispielsweise in einem Unternehmen gegeben, welches seine Mitarbeiter mit einem Rechner samt Internetzugang ausstattet.

Unmittelbarkeit

Unmittelbarkeit einer Abstimmung bedeutet, dass zwischen die Entscheidung

47 Siehe Schönau, Elektronische Demokratie, S. 88.

48 Zum Vergleich des gesamten Absatzes siehe Will, Internetwahlen, Kapitel 4.I, S. 75ff.

49 Dieser Aspekt findet sich beispielsweise bei Will, Internetwahlen. Schönau argumentiert dagegen, dass eine einhundertprozentige Sicherheit für keine einzelne Stimme besteht. Auch bei Präsenzwahlen kann es zum Verlust des Wahlscheines auf dem Postweg, einem Fehler im Wahlverzeichnis oder einem Verlust des Stimzettels führen.

unter dem formulierten Abstimmungsergebnis keine weitere Willensentscheidung treten darf. Im Falle einer Wahl meint dies die direkte Verwendung der Wahlstimmen für die Zuteilung der Abgeordnetensitze. Nicht jede Wahl fordert die Unmittelbarkeit.

Im Fall von elektronischen Hilfsmitteln bei einer Abstimmung besteht keine besondere Einschränkung im Bezug auf die Unmittelbarkeit dieser. Im Falle einer Internetwahl hängt die Unmittelbarkeit mit der sicheren Übertragung der Stimme zusammen.

Ein funktionierendes Internetabstimmungssystem muss also entsprechende Sicherungsmaßnahmen der Übertragung bereitstellen können, beispielsweise durch die Nutzung des SSL-Protokolls. Ohne in der Literatur erwähnt zu werden, besteht allerdings auch bei Nutzung von SSL die Gefahr eines „Man-In-The-Middle"-Angriffes[50]. Da mit einem solchen Angriff immer nur die Abgabe **einer** Stimme manipuliert werden kann, würde es zur Einflussnahme auf ein Mehrheitsergebnis allerdings erheblicher Aufwände bedürfen. Dies ist vergleichbar mit der Aneignung eines Wahlscheines und einer Unterschriftenfälschung im Wahllokal.

Freiheit

Der Grundsatz der Freiheit einer Abstimmung gebietet, das Stimmrecht ohne Zwang oder sonstige unzulässige Beeinflussung von außen ausgeübt werden kann. Hierin unterscheiden sich elektronische Wahlhilfen im Wahllokal nicht vom herkömmlichen Verfahren. Auch Wahlmaschinen müssen die Möglichkeit bieten ohne Beeinflussung (und Beobachtung) seine Stimme abzugeben. Dies sollte in der Regel zu gewährleisten sein.

Durch seinen Charakter als Distanzwahl erfährt eine Internetwahl die gleichen Einschränkungen wie etwa eine Briefwahl. Von staatlicher Seite sind die Stimmberechtigten nicht gegen die Einflußnahme Dritter im privaten Umfeld geschützt, die Freiheit der Wahl ist nicht ohne Zweifel sicherzustellen. Das Bundesverfassungsgericht toleriert diesen möglichen Eingriff der Freiheit unter Zurückstellung des Prinzips hinter dem Prinzip der Allgemeinheit einer Wahl. Analog zur Briefwahl wäre eine Internetwahl also nur im Falle der Verhinderung eine zulässige Alternative.

In Bezug auf eine Internetwahl werden weitere Hemmnisse wie der nicht vorhandene Schutz vor Wahlwerbung im Moment des Wahlaktes durch Browser-Banner und Pop-Ups genannt[51]. Dies bedeutet für ein Internetabstimmungssystem, dass es gänzlich frei von Werbung finanziert sein muss. Besonders bei Verwendung von konglomerierenden Werbepartnern[52] ist die Art der tatsächlich eingeblendeten Werbung nicht kontrollierbar.

---

50 Siehe dazu Wikipedia, Man-In-The-Middle-Angriff und weiterführende Links.

51 Zum Beispiel bei Will, Internetwahlen, S. 124ff.

52 Das System AdSense der Firma Google ist ein Beispiel dafür. Auf der Webseite wird ein Feld bereitgehalten, welches Werbepartner der Firma Google mit Werbeinhalten füllen. Dem Anbieter entfallen Aufwände für gezielte eigene Aquise von Werbung auf seiner Seite, er verliert aber auch Kontrollmöglichkeiten. Siehe www.google.com/adsense, zuletzt gesehen im März 2009.

Mit der Freiheit einer Abstimmung ist auch die Freiheit auf Verzicht gemeint. Dies bedeutet, dass ein Wähler das Recht hat, seine Stimme ungültig zu machen. Dies muss von einem Internetabstimmungssystem berücksichtigt werden. Gleichzeitig bietet ein elektronisches System einen erheblichen Schutz vor einer versehentlich ungültig abgegebenen Stimme[53].

Gleichheit

Die Gleichheit einer Abstimmung verbürgt, dass Jedermann sein Stimmrecht in formal möglichst gleicher Weise ausüben können soll. Elektronische Hilfsmittel im Wahllokal bedeuten keine Einschränkung der Gleichheit, wenn Sie flächendeckend eingesetzt werden. Im Falle eines Partikulareinsatzes müssen sich Beschriftungen im Wesentlichen mit dem manuellen Stimmzettel decken. Es reicht, wenn gleiche Elemente nach ihrem Zweck große Ähnlichkeit aufweisen[54]. Dies gilt im Hinblick auf das Layout – Schrift, Format, farbliche Gestaltung – als auch auf die Anordnung der Wahlvorschläge.

Im Falle der Internetwahl ergeben sich erneut Analogien zur Briefwahl. Es kann nicht grundsätzlich ausgeschlossen werden, dass ein Familienoberhaupt Stimmen für die ganze Familie abgibt und damit seine Stimme mehrfach gewertet wird. Im Falle der Briefwahl wird dies Problem durch die handschriftlich abzugebene Versicherung an Eides statt, den Stimmzettel persönlich ausgefüllt zu haben, gemindert. Im Falle einer vollständigen Digitalisierung des Abstimmungsvorganges könnte an diese Stelle eine digitale Signatur treten[55]. Darüber hinaus werden auch in Hinblick auf die Gleichheit der Wahl vermehrte Manipulationsmöglichkeiten während der Übertragung und durch die verarbeitende Software genannt. Diese müssen einerseits durch Sicherung der Übertragung (siehe Abschnitt zum Prinzip der Unmittelbarkeit) und andererseits durch eine Softwarevalidierung[56] verhindert werden.

Um überhaupt eine gewisse Form von Transparenz in der Verarbeitung der Stimmen durch Informationstechnologie auch für Laien zu gewährleisten ist nur die Verwendung von Open-Source-Software denkbar[57]. An die Stelle der im Wahllokal kontrollierenden Wahlhelfer tritt die breite Masse der Öffentlichkeit. Im

53 Bei der Bundestagswahl 2005 betrug der Anteil ungültiger Zweitstimmen 1,6% und der prozentuale Unterschied zwischen den beiden Volksparteien CDU und SPD lediglich 1%.

54 Siehe dazu Schönau, Elektronische Demokratie, S 93. Robort Schönau verweist dabei auf ein Urteil des Wahlprüfungsgerichtes des hessischen Landtages, Staatsanzeiger 1983, S. 1066f.

55 Hieran schließt sich schon lange währende Diskussion um die Gleichwertigkeit von Unterschriften und digitalen Signaturen an, siehe Skrobotz, Das elektronische Verwaltungsverfahren, S. 66ff.

56 Dies betrifft im Besonderen die Öffentlichkeit einer Wahl, einem sogenannten ungeschriebenen Wahlprinzip. Eine Verletzung dieses Prinzips wurde vom Bundesverfassungsgericht in seinem Urteil vom 3. März 2009 bei der Verwendung von Wahlmaschinen gerügt.

57 Siehe Schönau, Elektronische Demokratie, 2007, S. 97. Schönau erwähnt Open-Source als eine Möglichkeit zur Steigerung der Transparenz beim elektronischen Wählen. Als zweite Lösung nennt er ein System kontiniuierlicher Zertifizierung, wie dies bereits erfolgreich durch die Physikalisch-Technische-Bundesanstalt für bestehende Wahlcomputersysteme praktiziert werde. Das BVerfGE bewertet diese Art der Zertifizierung in seinem Urteil vom 3. März 2009 ausdrücklich als nicht ausreichend zur Wahrung des Wahlprinzips der Öffentlichkeit.

Idealfall erhält Jedermann lesenden Zugriff auf eine serverseitig betriebene Software. Im Gegenzug entstehen natürlich große Angriffspotentiale, die aus der genauen Kenntnis der Softwarearchitektur durch den Angreifer entstehen. Dieses Prinzip der vollständigen Offenheit kann der Kryptographie entlehnt werden[58]. Dort sorgen offene Verfahren auch für sichere Verfahren, weil Fehler schnell gefunden werden. Durch die Transparenz sorgen sie ebenfalls für vertrauenswürdige Verfahren.

Geheimheit

Die Geheimheit einer Abstimmung ist besonders im Hinblick auf Remote-Internet-Abstimmungen ein Problem, sie ist dagegen bei jeder Form des Einsatzes elektronischer Abstimmungen in einem Wahllokal weitgehend unproblematisch[59]. Hintergrund ist die Tatsache, dass im Fall eines Distanzwahlverfahrens die Identifikation der Abstimmungsberechtigung sichergestellt werden muss. Eine gleichgelagerte Problematik existiert analog im Falle der Briefwahl, die Internetwahl kann also an dieser Rechtmäßigkeit gemessen werden. Bei einer Briefwahl werden Wahlschein und Wahlzettel in getrennten Umschlägen versendet. Nach der Sicherstellung der Wahlberechtigung selbst wird der Umschlag mit dem Wahlzettel unter der Aufsicht der Öffentlichkeit[60] in die Wahlurne gelegt und werden die Stimmzettel anschließend ausgezählt. Um die dabei erfolgte Trennung von personenbezogenen und stimmbezogenen Daten in einem digitalen System abzubilden, wurden komplexe Verfahren entwickelt, sogenannte Verblendungssysteme[61]. Gemeinsam mit asymmetrischer Verschlüsselung zur sicheren Übertragung und digitalen Signaturen zur Identifikation ist es einem Internetabstimmungssystem technisch möglich, das Wahlprinzip der Geheimheit auf dem Niveau der Briefwahl abzubilden[62].

Die Briefwahl sichert durch die Abstimmungsoption auch für Wähler, die am Wahltag verhindert sind, die Allgemeinheit der Wahl. Die Briefwahl wurde vom Bundesverfassungsgericht zuletzt Anfang der Achzigerjahre für zulässig erklärt. Das Gericht ließ ausdrücklich den Grundsatz der Geheimheit einer Wahl vor dem Grundsazt der Allgemeinheit zurücktreten. Das bedeutet, dass eine Briefwahl nur im Ausnahmefall und auf Antrag zulässig ist[63]. Die Einführung von Internetwahlen würde keine weitere Stärkung der Allgemeinheit von Wahlen bewirken, weil die Möglichkeit der Briefwahl bereits besteht. Eine Internetwahl

58 Die Forderung, dass die Sicherheit eines Verschlüsselungsverfahrens auf der Geheimhaltung des Schlüssels und nicht auf der Geheimhaltung ds Verfahrens beruhe, wird Kerckhoffs'sches Prinzip genannt. Der Kryptologe Auguste Kerckhoffs formulierte diesen Grundsatz bereits 1883 in seinem Werk La cryptographie militaire. Siehe dazu Bruce Schneider, Applied Cryptography, 2. Ausgabe, 1996.

59 Siehe dazu Schönau, Elektronische Demokratie, S. 139 und Will, Internetwahlen, S. 133.

60 Nach dem Wahlprinzip der Öffentlichkeit hat jeder Wahlberechtigte die Möglichkeit die Stimmauszählung zu verfolgen, auch wenn dies Recht in der Praxis kaum genutzt wird. Die Kontrolle erfolgt das Mehraugenprinzip der anwesenden Wahlhelfer.

61 Siehe Will, Internetwahlen, S. 147ff; Schönau, Elektronische Demokratie, S. 155ff und Karpen, Elektronische Wahlen? - Einige verfassungsrechtliche Fragen, S. 29ff.

62 Karpen [P19] spricht von Verfahren, die insgesamt den Eindruck von Verlässlichkeit vermitteln, Schönau weist darüber hinaus auf die Notwendigkeit von praktischen Versuchen zum Beweis der tatsächlich hinreichenden Sicherheit hin.

63 Siehe Karpen, Elektronische Wahlen? - Einige verfassungsrechtliche Fragen, S. 28f.

wäre der Briefwahl allerdings insoweit überlegen, dass die Verfahren sicherer sind[64] und der Wahlakt zeitgleich mit den Präsenzwahlen stattfindet und nicht wie bei der Briefwahl mit einem Verzug. Dem steht die mangelnde Transparenz der eingesetzten Verfahren gegenüber.

Zwischenergebnis

Die Geheimheit von elektronischen Wahlsystemen im Wahllokal steht nicht Frage. Im Einsatz von Internetwahlen sind komplexe Verfahren erforderlich, für die bisher keine Implementierung vorliegt. Internetwahlen können in ihrem Charakter als Distanzwahl prinzipbedingt nur eine Briefwahl ersetzen.

Gremienabstimmungen

Gremienabstimmung per Internet werden in der Literatur bisher nicht diskutiert. Es ist offenbar geworden, dass eine Einhaltung des Prinzip der Geheimheit (Trennung von personen- und stimmbezogenen Daten) auf dem Niveau einer Bundestagswahl bei Gremienabstimmungen nicht oder nur sehr aufwendig möglich ist. Eine Alternative ist die gezielte Inkaufnahme der digitalen aber nicht transparenten Trennung von personenbezogenen und stimmbezogenen Daten. Eine Voraussetzung für den Einsatz eines Internetabstimmungssystems wäre die explizite oder implizite Festschreibung eines solchen Verzichts in der Satzung oder Geschäftsordnung des betroffenen Gremiums. In einigen Teilen ist eine solche Vorgehensweise bereits heute Realität. Handschriftlich ausgefüllte Stimmzettel bei Vorstandswahlen mit Vorschlagsrecht schließen beispielsweise eine Rückverfolgung durch Skriptologen (Handschrifterkennung) nicht sicher aus. Eine weitere Möglichkeit ist, auf die Erfassung von personenbezogenen Daten gänzlich zu verzichten und schwächere Identifikationsmöglichkeiten beispielsweise in Form eines Zugangscodes zu erlauben. Dieser wäre nicht gegen die Weitergabe an Dritte geschützt.

Ungeschriebene Wahlgrundsätze

Außer den in Artikel 38 des Grundgesetzes ausdrücklich genannten Wahlrechtsgrundsätzen gibt es weitere sogenannte ungeschriebene Grundsätze. Dazu werden die Öffentlichkeit der Wahl, ihre Gleichzeitigkeit, die Verständlichkeit gezählt und darüber hinaus die Einfachheit, Verlässlichkeit und Kostenfreiheit des Wahlvorganges.

Der Grundsatz der Öffentlichkeit bezieht sich dabei auf jede Phase der Wahl. Nach Karpen [P19] kann eine Internetwahl schon deshalb „nur eine Ausnahme vom öffentlichen Votum im Wahllokal“ sein, denn die „Onlinewahl ... kommt einer Entführung der politischen Wahl aus dem öffentlichen Raum“ gleich. Im Falle von Wahlcomputern sind besondere Vorkehrungen zu treffen, um die „Öffentlichkeit der Wahlvorbereitung, -durchführung, und ergebnisauszählung nicht erodieren“ zu lassen. Damit ist sowohl eine Transparenz über die Stimmabgabe, als auch Transparenz in der Rekonstruierbarkeit des Zustandekommens des Wahlergebnisses gemeint.

Ebenso müsste der Wahlgrundsatz der Einfachheit bei der Benutzung der neuen

64 Siehe Schönau, Elektronische Demokratie, S. 148.

Technik besonders berücksichtigt werden. Im Falle von digitalen Systemen besteht immer die Gefahr eines technischen Defektes. Dies macht die Bereithaltung eines Alternativplans auch unter Rückgriff auf das herkömmliche Wahlzettelsystem bei elektronischer Wahlen notwendig, um dem ungeschriebenen Wahlrechtsgrundsatz der Verlässlichkeit zu entsprechen.

Im Falle einer ausschließlichen Internetwahl müssen die Kosten für einen Rechner oder mögliche Chipkartenlesegeräte zur Identifikation vom Wahlveranstalter zur Verfügung gestellt werden, es sei denn eine Chipkarte zur allgemeinen Identifikation im Rechtsverkehr und die Nutzung des Computers gehörten – in der Zukunft – zur allgemeinen Lebensführung.

4.2 Bundesverfassungsgericht

Nicht nur weil es in die Bearbeitungszeit dieser Arbeit fällt, ist ein Urteil des Bundesverfassungsgerichtes[65] interessant. Darin wird der Betrieb der bei der Wahl zum 16. Deutschen Bundestag in einigen Gemeinden eingesetzten elektronischen Wahlgeräte als unvereinbar mit dem Grundgesetz befunden. Dies gilt ebenso für die dem Einsatz der Geräte zugrundeliegende Bundeswahlgeräteverordnung. Der Grund liegt in der Verletzung des verfassungsrechtlichen Grundsatzes der Öffentlichkeit einer Wahl, für welches die betroffenen Geräte keine ausreichende Gewährleistung liefern können.

Nach der Klassifikation aus Kapitel III.2 handelt es sich also um Wahlcomputer. Diese wurden von einem privaten Unternehmen konzipiert und der Physikalisch-Technischen Bundesanstalt (PTB) erfolgreich zur Prüfung vorgelegt. Das Bundesverfassungsgericht rügt nun in seinem Urteil, dass die im System verwendete Software nicht öffentlich zugänglich ist – sie liegt auf einem EPROM. Damit ist auch das Zustandekommen des Wahlergebnisses nicht für Jedermann ersichtlich. Die Prüfung durch die PTB ist nicht ausreichend, weil nicht ohne Zweifel sichergestellt werden kann, dass die von der PTB geprüfte Softwareversion mit der tatsächlich eingesetzten identisch ist[66].

Interessant ist, dass genau dieses Prüfverfahren von Schönau in [P14] noch im Jahr 2007 aus ausreichend bewertet wurde[67]. Der klageführende Prof. Dr. Ulrich Karpen legt seine Bedenken in Bezug auf die Öffentlichkeit einer Wahl jedoch schon 2005 in [P19] dar. Letztlich ist mit dem Urteil ein Präzidenzfall und eine Richtungsentscheidung gefallen, die einige Widersprüche unterschiedlicher Rechtsauffassungen auflöst. Im Urteil wird aber auch klargestellt, dass der Einsatz von elektronischen Wahlen und insbesondere Wahlcomputern keinesfalls grundsätzlich verfassungswidrig wäre[68]. Lediglich die Wahlrechtsgrundsätze müssen eingehalten werden. Dies bedeutet für einen Wahlcomputer oder gar ein webbasierte Abstimmungssystem im Hinblick auf den Wahlrechtsgrundsatz der Öffentlichkeit, dass es quelloffen sein sollte.

65 BVerfG, 2 BvC 3/07 vom 3. März 2009, Absatz-Nr. (1-163)
66 Siehe BVerfG, 2 BvC 3/07, Absatz-Nr. 123
67 Siehe Schönau, Elektronische Demokratie, S. 55f.
68 Siehe BVerfG, 2 BvC 3/07, Absatz-Nr. 121

4.3 Quintessenz

Alle Autoren erkennen in der elektronischen Durchführung von Wahlen große Potentiale und versprechen sich ähnliche Aktivierungspotentiale wie in anderen Elementen der elektronischen Demokratie[69]. Gleichzeitig wird der Rückzug von demokratischen Entscheidungen aus dem öffentlichen Raum auch als Problem erkannt. Die Verfassungsgemäßheit ist nur unter großen Anstrengungen sicherzustellen – wenn überhaupt. Will [P12] betrachtet die Durchführung von Internetwahlen als unzulässig, zumindest im aktuellen Stand der Technik. Schönau [P14] stellt dagegen fest, dass „elektronisches Wählen mit elektronischen Stimmenzählgeräten und als Onlinewahl verfassungsrechtlich zulässig" sind. Übergreifend ist festzuhalten, dass die Verfassungsgemäßheit nur ein gedankliches Konstrukt ist, es gibt bisher kein fertiges System, welches die beschriebenen Anforderungen erfüllen würde. In diesem Sinne hat auch das Bundesverfassungsgericht die Nutzung der niederländischen Wahlcomputer als nicht verfassungsgemäß abgelehnt, nicht jedoch den Einsatz von Wahlcomputern grundsätzlich ausgeschlossen. Die Umsetzung eines Systems beschriebener Güte bedarf daher noch einiger Zeit. Autoren schlagen daher – wenn elektronische Wahlen denn eingeführt werden sollen – mehrheitlich ein mehrstufiges System vor, welches zunächst Zählmaschinen in Wahllokalen, später vernetzte Wahllokale und schließlich Wahlen auch außerhalb von Wahllokalen vorsieht[70]. Diese Maßnahmen sind vor dem Hintergrund entstehender Kosten zu reflektieren.

Die Betrachtungen finden vor dem Hintergrund der Wahlprinzipien formuliert im Grundgesetz und damit nach der Rechtssprechung des Bundesverfassungsgerichtes statt. Für weniger relevante Rechtsräume wären möglicherweise auch weniger strenge Anforderungen denkbar. Hier könnten hinter das Prinzip der Allgemeinheit (möglichst gute Zugänglichkeit) andere Prinzipien weiter zurücktreten. So wäre eine Internetwahl für Gremien denkbar, deren Mitglieder über weit verteilte Wohnsitze verfügen[71]. Dem offensichtlichen Mehrwert stünden dann Einschnitte wie beschrieben gegenüber. Möglicherweise ein sinnvoller Kompromiß.

Ein solcher Kompromiß macht vor allem bei der Übertragung der Überlegungen von Wahlen auf Abstimmungen Sinn. Zum einen können Gremien durch eigene Satzungen oder Geschäftsordnungen eigene Abstimmungsverfahren beschließen. Dies reduziert die rechtliche Diskussion. Zum anderen liegt aufgrund der geringeren Tragweite einer Abstimmung im Gegensatz zu einer Wahl bereits grundsätzlich weniger Brisanz im Einsatz eines internetbasierten Systems. Das Potential in der Überbrückung von zeitlichen und räumlichen Distanzen beim Einsatz eines internetbasierten Systems „im Kleinen" scheinen die verfassungsrechtlichen Einschränkungen aufzuwiegen. Weiterhin wirken einige

69 Siehe Schönau, Elektronische Demokratie, S. 229, Punkt 3.

70 Siehe Khorrami, Bundestagswahlen im Internet – Zur rechtlichen und tatsächlichen Realisierbarkeit von Internetwahlen, S. 196f; Karpen, Elektronische Wahlen? - Einige verfassungsrechtliche Fragen, S. 60f; Will, Internetwahlen, S. 161f und eingeschränkt Schönau, Elektronische Demokratie, S. 236ff.

71 Ein Beispiel sind die Wahlen zum „Board of Directors" der wichtigen Internet Corporation for Assigned Names and Numbers (ICANN), welches unter anderem über die Vergabe von Top-Level-Domains entscheidet. Das Board besteht aus 21 Mitgliedern aus aller Welt, die per Internetwahl gewählt werden. Siehe dazu auch Will, Internetwahlen, S. 63ff.

Einwände gegen den Einsatz von internetbasierten Systemen akademisch. Mögen sie in Bezug auf den Stellenwert einer Bundestagswahl Berechtigung haben, so können an Personalratswahlen, Stupawahlen und Abstimmungen pragmatischere Maßstäbe angelegt werden.

Letztlich ist ein webbasierte Abstimmungssystem mit gewissen Sicherheitsmechanismen zunächst nur ein Zusatzangebot. Sein Einsatz läge im jeweiligen Ermessen des betroffenen Gremiums. Mit der Existenz eines einsatzbereiten Systemes besteht jedoch darüber hinaus die berechtigte Hoffnung auf Abstimmungen, an deren Durchführung ohne ein verfügbares System nicht gedacht worden wäre. Damit kann die Onlinelösung einen Beitrag zur Verbesserung der gesellschaftlichen Demokratisierung leisten.

5. Soziologische Aspekte

Die Auswirkungen von technischer Verbesserungen durch das Internet insbesondere im Hinblick auf neue Möglichkeiten der Kommunikation ist auch ein Untersuchungsgegenstand der Soziologie. Das Stcichwort *Informatik und Gesellschaft* ist dabei ein Querschnittsthema, welches auch von Informatiklehrstühlen beleuchtet wird. Die *Gesellschaft für Informatik (GI)* hat sich das Thema der Elektronischen Demokratie ebenso zu eigen gemacht. In der soziologischen Literatur zu dem Thema wird die digitale Kommunikation demokratietheoretisch untersucht. Dies geschieht vor dem Hintergrund einiger optimistischer Aussagen verantwortlicher Politiker einerseits und Vertretern der informationstechnischen Industrie andererseits. So beschrieb bereits 2001 der damalige Bundesinnenminister und Jurist Otto Schily folgende Vision:

Durch Diskussionsforen im Internet, Live-Chats mit Politikern oder E-Mails an die Regierung ist es möglich, in der Demokratie des digitalen Zeitalters die Agora, also den politischen Marktplatz der alten Athener, als E-Gora wieder attraktiv zu machen.[72]

Die Wissenschaft beschreibt zwei Wege, den die Zukunft gehen könnte, um diesen demokratietheoretischen Idealzustand zu erreichen[73]. Der erste Weg ist das sogenannte *Net-Empowering*. Diese Richtung geht davon aus, dass das Internet wegen seiner revolutionären Kommunikations- und Informationsmöglichkeiten das politische Bewußtsein der Gesellschaft steigert. Dies führt zu einer verstärkten Teilnahme am politischen Prozess. Das Internet relativiert räumliche und institutionelle Distanz und schafft damit Anzeize für Volksvertreter, ihr Wissen über die tatsächlichen Interessen ihrer Wähler zu verbessern und in ihren Entscheidungen stärker zu berücksichtigen. Durch den vermehrten Einsatz von Volksentscheiden wird die repräsentative Demokratie schließlich einer plebiszitären Demokratie attischem Vorbilds immer ähnlicher.

Die zweite Richtung betrachtet demgegenüber nicht die Interaktion zwischen Volk und Volksvertretung, sondern zwischen Bürgern untereinander[74]. Das Internet fördert demnach den Austausch von Bürgergemeinschaften und dadurch lassen sich kommunale und lokale Probleme vor Ort und dennoch dezentral lösen. Räumlich voneinander getrennte Gemeinden finden eine Unterstützung, als ginge es um unmittelbare Nachbarschaftshilfe. Schließlich entstehen nach diesem Modell virtuelle Bürgerschaften, die im und über das Netz sozial und politisch handeln. Mit dieser vermehrten politischen Partizipation auf kommunaler Ebene und der Verknüpfung über räumliche Entfernungen hinweg würde damit aus der repräsentativen Demokratie mehr und mehr eine direkte Demokratie werden, also ein Regieren durch (virtuelle) Versammlungen und Abstimmungen der

72 Siehe Ausgabe der Frankfurter Allgemeine Zeitung vom 20. März 2001, S. B13.

73 Siehe Siedschlag/Bilgeri/Lamatsch, Kursbuch Internet und Politik – Elektronische Demokratie und virtuelles Regieren, Band 1/2001, S. 10f.

74 Siehe auch Abschnitt III.1 und dem dortigen Schema E-Government der Definition von Speyer.

Bürgerschaft selbst.

Die Entwicklung dieser Vision der Basisdemokratie wird jedoch keinesfalls nur positiv bewertet. So stellt sich die Frage, ob durch die öffentliche Meinung direkt getriebene Entscheidungen der Volksvertreter in Summe überhaupt dem Gemeinwohl dienen und damit nicht sogar notwendige Sachentscheidungen verhindert werden. Weiterhin werden mögliche Gefahren genannt, wie verantwortungsloses Protestverhalten und mangelnde demokratische Stabilität durch den Legitimitätsverlust der aus jahrhundertelanger geschichtlicher Erfahrung hervorgegangenen Institutionen[75].

Einige Autoren beurteilen die Vision der E-Gora gar als illusiorisch. Denn eine der wesentlichen Eigenschaften der Agora und des römischen Forums ist die Öffentlichkeit. Nur durch den Austritt aus der Anonymität und der Abhebung „von dem Geraune, dem Geplapper [der antiken Bürger] auf der Agora“ begründet sich der Wert des demokratischen Prinzips. Gerade dieser Gedanke stünde im Gegensatz zur anonymen Philosophie des Internets. Dieses als öffentlichen Raum zu bezeichnen, unterstellt demnach eine Transparenz und Überschaubarkeit, die nicht vorhanden und auch nicht realisierbar ist[76].

Insgesamt gehen die Betrachtungen jedoch an der Idee eines webbasierten Abstimmungssystems vorbei. Ob und inwieweit elektronischen Wahlen und der Einfluss der digitalen Medien das Wesen der Gesellschaft an sich beeinflussen werden, scheint offen. Die technische und rechtliche Realisierbarkeit mancher Utopie ist nicht zwingend gegeben. Ein Abstimmungssystem dagegen muss als Zusatzangebot verstanden werden. In definierten Einzelfällen mit zeitlich engen oder räumlich distanzierten Rahmenbedingungen ist es in jedem Fall eine Bereicherung. Gremien können eigene weitere Rahmenbedingungen für den Einsatz eines solchen Systems schaffen. Daraus ist aber sicherlich nicht ein Automatismus für die Ablösung bisheriger Wahlverfahren oder gar die Umwandlung der repräsentativen zu einer direkten Demokratie zu folgern.

75 Siehe Siedschlag/Bilgeri/Lamatsch, Kursbuch Internet und Politik – Elektronische Demokratie und virtuelles Regieren, Band 1/2001, S. 11.

76 Siehe Gellner, Das Internet: Digitale Agora oder Marktplatz der Eitelkeiten in Koziol/Hunold, E-Demokratie = Ende der Demokratie, S. 12ff.

6. Einsatzbeispiele

Vor dem Hintergrund der theoretischen Betrachtungen macht eine Bestandsaufnahme existierender Systeme Sinn. Es fällt auf, dass die Betrachtungen der rechtlichen und technischen Anforderungen in der Literatur häufig theoretische Modelle bleiben. Einige Autoren schildern zwar auch den Praxiseinsatz von elektronischen Wahlen. Diese befassen sich häufig mit einem Randbereich von Wahlen zur Erprobung, beispielsweise Wahlen zum Studierendenparlament, oder Sozialwahlen. Die eingesetzen Systeme sind vielfältig, aber es ist kein einheitlicher Standard erkennbar, die Systeme verfolgen unterschiedliche Ziele und haben unterschiedliche Architekturen. Eine komplette Darstellung würde über den Rahmen dieser Arbeit hinausgehen.

6.1 Wahlsysteme

Interessanter sind verfügbare *Internet-Remote-Voting*-Systeme, die eine private Internetwahl ermöglichen[77]. Khorrami [P20] nennt Details zu einigen bekannten Verfahren. Für Deutschland ist das *i-vote* genannte System der von der Bundesregierung bereits im Jahr 2000 eingesetzen Forschungsgruppe Internetwahlen [O24] zu nennen. Es ist beispielsweise bei den oft genannten Wahlen zum Studierendenparlament der Universität Osnabrück im Jahr 2001 zum Einsatz gekommen. Das System zeichnet sich durch die Trennung der sich gegenseitig kontrollierenden Instanzen eines Zertifikatsausstellers, einem Wahlamtserver und einem Wahlurnenserver aus. Es wird damit versucht das Melderegister, das Wahlregister und die Wahlurne einer realen Wahl virtuell nachzubilden. Die Architektur unterscheidet sich insofern von einem rein webbasierten Abstimmungssystem, als dass der Wahlwillige im Vorfeld der Wahl eine persönliche und ausweisliche Registrierung vornehmen muss. Die Authorisierung erfolgt am Wahltag durch eine Chipkarte, welche ein Kartenlesegerät am heimischen PC erforderlich macht. Die eigentliche Übertragung der Stimme erfolgt jedoch nicht verschlüsselt.

Das *Platinum-Election-System (PES)* der Firma *VoteHere.Net* dagegen verwendet ein komplexes *Public-Key*-Verfahren, welches jedoch proprietär und nicht transparent kontrollierbar ist. Für den Zugang zur Wahl ist ebenfalls eine Registierung notwendig, die jedoch auf digitalem Wege und nur durch Angabe von persönlichen Daten wie Name und Geburtsdatum erfolgt. *PES* wurde beispielsweise für Testnominierungswahlen der Demokratischen Partei in den USA eingesetzt.

Nah an einem rein webbasierten System ist ein Angebot der Firma *SafeVote* [O25]. Der Wähler erhält lediglich seine Wahlunterlagen auf dem Postweg. Diese enthalten ein persönliches Zertifikat zur Authorisierung des Wahlberechtigten und einen sogenannten *Return-Code* für die konkrete Stimmabgabe. Diese erfolgt webbasiert über eine per *SSL* geschützte Verbindung.

77 Zur Klassifikation siehe Abschnitt III.2.

Die genannten Systeme stellen nur eine kleine Auswahl dar. Im Kern erfolgt die Beurteilung der Verfahren jedoch meist am Maßstab einer (Personen-) Wahl. Die Analysen sind zu abstrakt, um den konkreten Aufbau der Webseite zu beschreiben. Weiterhin sind keine Systeme bekannt, die ihre Architektur oder gar den Quelltext offen gelegt hätten.

6.2 Umfragesysteme

Es existieren dagegen auf dem deutschen Markt viele ausgereifte Umfragesysteme. Diese bieten insbesondere Firmen maßgeschneiderte Angebote für anonymisierte Mitarbeiterbefragungen. Im Gegensatz zur Wahlsoftware gibt es auch einige kostenlose Angebote[78]. Auf eine komplette Darstellung wird an dieser Stelle verzichtet, weil Abstimmungen und nicht Umfragen Hauptgegenstand der Untersuchung sind.

6.3 Polyas

Das interessanteste Angebot zum Thema webbasierte Abstimmungen stammt von der Firma Polyas [O29]. Polyas bietet neben klassischer Wahlsoftware zur Durchführung von privaten Internetwahlen auch ein *Polyas Abstimmung* genanntes System zur Durchführung von Abstimmungen an. Zielgruppe sind dabei nach Unternehmensangaben „Organisatinen, Vereine und Unternehmen“, die häufig verschiedene Formen von Abstimmungen durchführen müssen. Das System ist proprietär und nicht kostenlos[79], es existiert aber eine kostenfreie Demoversion für bis zu fünf Stimmberechtigte. Die Wahl des Präsidiums der Gesellschaft für Informatik e.V. werden beispielsweise mit dem Polyas-System durchgeführt.

Stellungnahme: Ich bin während der Erstellung dieser Arbeit erst sehr spät auf das Angebot der Firma Polyas aufmerksam geworden. Sonst wäre ein Vergleich und eine Orientierung des Systems ***Rogotio.*****net** am Polyas-System eine gute Möglichkeit gewesen. Nachdem bereits wesentliche Architekturentscheidungen gefallen waren, machte dies aus meiner Sicht keinen Sinn mehr.

78 Siehe beispielsweise das Fragebogen-Tool [O27] oder Vote24 [O28].

79 Die Kosten belaufen sich bis 20 Stimmberechtigte auf EUR 194,95, bis 100 Stimmberechtigte auf EUR 645,95 und bis 500 Stimmberechtigte auf EUR 1745,95 pro Jahr (Stand März 2009).

Teil IV Softwaretheorie

Der zweite Teil der Arbeit gibt eine Einführung in die zur Erstellung einer Webanwendung notwendige Theorie von Software-Engineering-Prozessen. Insbesondere erfolgt die Bereitstellung der Theorie des Rational-Unified-Process, welche im Zuge des Projektes erarbeitet worden ist.

1. Stand der Technik

Die Konzeption einer Online-Applikation kann auf vielfältigen Standards, Sprachen und Versionen aufsetzen. Dieses Kapitel beleuchtet kurz die verfügbare Technik.

1.1 HTML

Auch mit der rasenten Entwicklung des Internets ist die Grundlage jeder Webseite noch immer HTML. Diese Grundlage wurde um einige Elemente wie CSS erweitert und HTML wurde als XHTML standardisiert. Die überwiegende Zahl der am Markt befindlichen Browser können in ihren aktuellen Versionen den Standard korrekt interpretieren und darstellen.

Die Entwicklung von HTML als Seitenbeschreibungssprache, die in erster Linie Text darstellen sollte, kann auch heute nicht verleugnet werden. Dies zeigt sich beispielsweise bei der Gestaltung von Navigationsmenüs. Sie müssen aus einfachen Links und Buttons bestehen. Ausgefeiltere Navigationsmethoden erfordern mehr Aufwand in der Programmierung. Ein Beispiel dafür ist die Realisierung einer Baumstruktur oder einer sortierbaren Tabelle, deren Benutzung in in Rich-Clients Standardelemente sind.

Diese Tatsachen sind für den Programmierer einer Webseite Vor- und Nachteil zugleich. Sie bedeuten einen erheblichen Mehraufwand möchte man einen adäquaten Komfort zur Verfügung stellen. Gleichzeitig erlauben sie aber auch ungewöhnliche und freiere Gestaltungsmöglichkeiten. Man sieht es nativer Standardsoftware an, dass sie die immer gleichen Komponenten der bekannten Entwicklungstools einsetzen und nur selten vom ausgetretenen Pfad abweichen.

Letztlich kann man die Situation nicht ändern, sondern muss sich über die besonderen Anforderungen an gutes Webdesign aufgrund der eingeschränkten Voraussetzungen bewußt sein. Insbesondere ändern auch aktuelle serverseitige Techniken und Möglichkeiten der dynamischen Veränderung einer Seite nichts daran, dass die Struktur selbst in HTML wiedergegeben werden muss.

1.2 CSS

Mit der Verbreitung von HTML als Mark-Up-Sprache stiegen die Anforderungen an die farbliche Gestaltung und Aufbereitung von Text und anderen dargestellten Elementen. HTML selbst bietet dazu nur sehr eingeschränkte Möglichkeiten. Dabei konnte noch vor wenigen Jahren nicht von der korrekten Darstellung der CSS-Vorgaben durch die verbreiteten Browser ausgegangen werden, auch wenn die Codierung standardkonform war. Der bekannte Box-Bug[80] des Internet

80 Siehe SelfHTML [O03] unter dem Stichwort Boxmodell und Internet Explorer, http://de.selfhtml.org/css/formate/box_modell.htm.

Explorers ist ein Beispiel dafür. Stand heute wird der wesentliche Kern der CSS-Spezifikation weitgehend unterstützt. Anstoss dazu lieferte sicherlich die Verbreitung von Mozilla Firefox.

1.3 JavaScript

Der Sprachkern von *JavaScript* wurde von der Firma Netscape entwickelt und erstmal im Produkt Navigator 2.0 eingesetzt[81]. Heute existieren verschiedene Versionen und teilweise abweichende Bezeichnungen[82]. Die aktuelle Version ist *JavaScript* 1.8, welche jedoch nur im Mozilla implementiert ist[83] Der Internet-Explorer implementiert neben einigen weiteren Skriptsprachen, die Sprache *JScript*, welche in der Version 5.5 weitgehend kompatibel zu JavaScript 1.5 ist. Eine standardisierte Variante von *JavaScript* ist *ECMAScript*. Zur aktuellen Version ECMA v3 sind JavaScript 1.5 und JScript 5.5 konforme Implementierungen. Diese Darstellung betrifft clientseitiges *JavaScript*. Prinzipiell kann die Sprache auch in anderen Gebieten zum Einsatz kommen[84].

JavaScript ist eine umstrittene Technologien im Webkonkext. Dies hat dazu geführt, dass der Einsatz von *JavaScript* in allen Browsern deaktivert werden kann, wovon allem Anschein nach auch Gebrauch gemacht wird. Und in der Tat gibt es einige Argumente, die zunächst gegen den Einsatz von *JavaScript* zu sprechen scheinen.

1. *JavaScript* ermöglicht dynamische Veränderungen an einer Webseite und einigen Elementen der Browserumgebung, wie der Fenstergröße, sowie das Öffnen von neuen Fenstern. Dies wird von einigen Webdesignern für eine besonders eindringliche Darstellung ihres Angebots genutzt. So wird das Fenster auf volle Größe gezogen oder beim Schließen eines Fensters öffnet sich wieder ein neues Fenster. Nutzer fühlen sich von solchem Verhalten genervt.

2. Vordringlich im Internet Explorer, aber auch in anderen Browsern sind in der Vergangenheit Sicherheitslücken in der Implementierung der mit *JavaScript*-Ausführungsschicht bekannt geworden. Leider gibt es für einige der Sicherheitslücken keine Patches. Einziges Mittel bleibt dann die Abschaltung von *JavaScript*.

3. Das „Ebay-Problem“. Da *JavaScript* an beliebiger Stelle im Quelltext stehen kann und trotzdem interpretiert wird, kann es auch mit den Produktbeschreibungen des im Internet-Auktionshauses *Ebay* hochgeladen

81 Der Einsatz erfolgte zunächst unter dem Namen LiveScript, die Umbenennung in JavaScript war eine reine Marketingmaßnahme. JavaScript hat zunächst keinen Zusammenhang zur Sprache Java. Zur Abgrenzung siehe Flanagan, JavaScript und [W11].

82 Siehe dazu erschöpfend Flanagan, JavaScript – Die umfassende Referenz, 2. Auflage, S. 2ff.

83 Stand Dezember 2007. JavaScript 1.8 ist in Mozilla-Firefox seit der Version 3.0b2 implementiert.

84 Bereits existierende Beispiele sind der Windows Scripting Host, der JScript als Sprache unterstützt oder der IIS-Webserver von Microsoft. Der JavaScript-Interpreter von Netscape wurde wie der Navigator als Open-Source freigegeben. Siehe dazu Koch, JavaScript, 3. Auflage.

werden. Damit ist es dann möglich, Zugriff auf die Elemente der gesamten Seite zu nehmen, insbesondere auf die Referenz der Verweise zum Bieten. Diese lassen sich so auf eine private Seite umlenken, um dort die Eingabe des *Ebay*-Passwortes abzufangen.

Quintessenz: Nicht JavaScript selbst richtet Schaden an. Es ist vielmehr der Umgang mit der Technik seitens der Browserhersteller und einiger Webseitenbetreiber. Wenn eine Seite mit Ihrem dynamischen Verhalten nervt, so wäre es einfacher diese Seite nicht mehr zu betreten oder *JavaScript* für diese Seite selektiv abzuschalten. Sicherheitslücken – also Implementierungsfehler – existieren wohl in jeden System, gleichwohl scheint das Microsoft-Produkt etwas anfälliger zu sein. Hier bewahrheitet sich erneut die Theorie der sichereren quelloffenen Systeme, welche eine Fehlerkontrolle von Jedermann möglich machen.

Aus dem *Ebay*-Problem lässt sich folgern, dass Daten im Upload eine besonderen Sicherheitsprüfung unterliegen sollten. Dies gilt zum einen für *JavaScript*, aber beispielsweise auch sogenannte *SQL-Injections*, die wirksam abgefangen werden müssen. Das *Ebay*-Problem lässt sich im Übrigen mit einer selektiven Abschaltung von *JavaScript* lösen, eine empfehlenswerte Maßnahme.

Bei der Konzeption von *JavaScript* war ein wesentliches Kriterium die Beachtung von Sicherheitskriterien. Aus dem Grund kann clientseitiges JavaScript keine lokalen Dateien lesen oder schreiben und keinerlei Netzwerkoperationen durchführen. *JavaScript* darf dabei nicht mit proprietären Techniken für *ActiveX* oder das um weitere Möglichkeiten erweiterte *JScript* verwechselt werden. JScript zum Beispiel wird vom Internet-Explorer eingesetzt, um andere Technologien wie *ActiveX-Controls* zu steuern. Da diese durchaus weitreichendere Möglichkeiten und insbesondere Dateizugriff haben, ist Obacht geboten[85].

1.4 DOM

Das *Document Object Model* ist das Modell mit dem die Struktur einer HTML-Seite intern repräsentiert werden kann. Mit Hilfe von *JavaScript* kann es manipuliert werden.

1.5 AJAX

AJAX ist ein Akronym und steht für *Asynchronous JavaScript and XML*. Es bezeichnet ein Konzept der asynchronen Datenübertragung

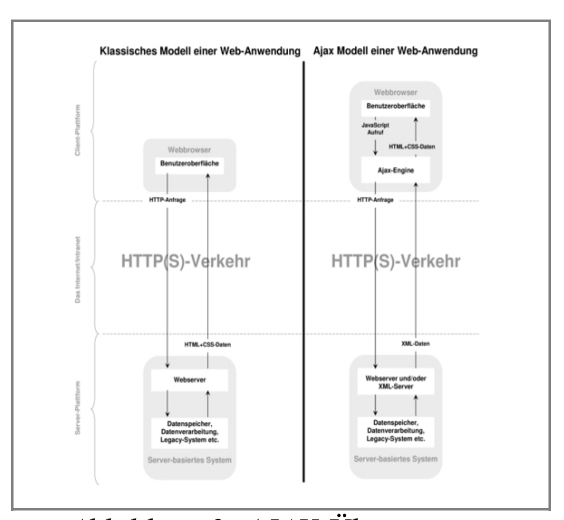


Abbildung 2: AJAX-Übertragung

85 Siehe dazu Flanagan, JavaScript – Die umfassende Referenz, Kapitel 21 – Sicherheit in JavaScript, S. 449ff.

zwischen einem Server und dem Browser, welches es ermöglicht, innerhalb einer HTML-Seite dynamisch eine HTTP-Anfrage auszuführen, ohne auf einen benutzererzeugten Request zu warten. Auf diese Weise können Teile der Homepage mit Serverdaten aktualisiert werden. Die Verbreitung von *AJAX* hat vor einigen Jahren für Aufsehen gesorgt, weil einige Anbieter die Technik so erfolgreich einsetzten, dass sie nativer Software glich. Ohne die Aktivierung von *JavaScript* ist Kommunikation über *AJAX* unmöglich.

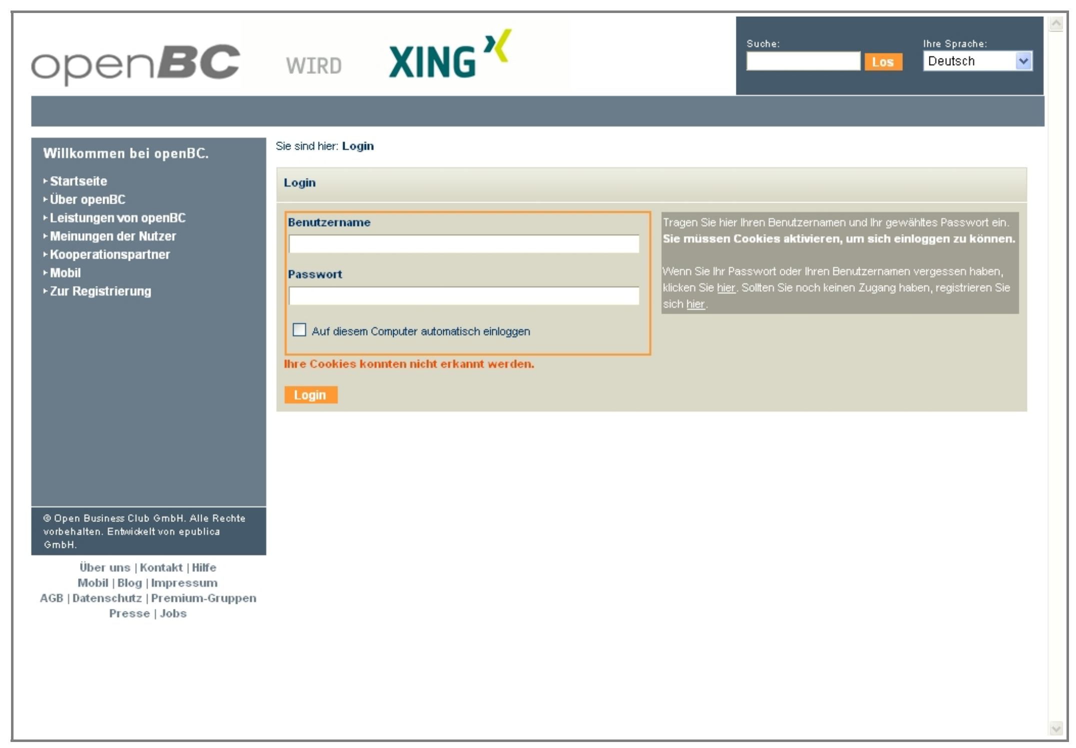


Abbildung 3: Deutlicher Hinweis auf den Einsatz von Cookies

1.6 Cookies

Der Einsatz von Cookies ermöglicht die Speicherung von Benutzerdaten über getrennte Seitenbesuche hinweg. Sie können zur Speicherung einer *Session-ID* genutzt werden, um so eingeloggt zu bleiben. Puristen neigen dazu, *fremde* Daten auf ihrem System abzulehnen und den Einsatz von Cookies zu untersagen. Sollte eine Seite Cookies verwenden wollen, so sollte sie dies explizit sagen. Ein gutes Beispiel dazu zeigt die Abbildung. Ideal ist der Einsatz von Cookies für zusätzlichen Konfort. Sind Cookies aktiviert, werden individuelle Einstellungen darin gespeichert, sind Cookies nicht aktiviert, muss der Benutzer auf diesen Service verzichten.

1.7 PHP

Die serverseitige Sprache PHP ist frei und sehr gut dokumentiert. Sie ist von jeher gut auf die Zusammenarbeit mit MySQL-Datenbanken abgestimmt und kann auf

einfachste Weise mit HTML-Ausgaben produzieren. Mit PHP 5 wurde die Sprache um eine vollwertige Objektorientierung erweitert.

1.8 Technologiekombination

Eine Schwierigkeit in der Anwendung der genannten Techniken ist ihre Kombination im Quelltext. Folgendes Listing zeigt eine nicht abwägige Situation.

```
$raw = mysql_query(„SELECT * FROM 'table'");
$data = mysql_fetch_assoc($raw);
print(„<div style='color:#000' onmouseover='doit(self,
\"X\");'>");
print($data[column]);
print(„</div>");
```

Alleine schon bei der korrekten Benutzung der Anführungszeichen treten Probleme auf. Die Kombination von SQL-Statements, PHP-Anweisungen, HTML-Markup und darin wiederum *JavaScript*-Anweisungen sollte sauber getrennt werden, um Übersichtlichkeit zu bewahren.

2. Design und Usability

Eine Webanwendung stellt besondere Anforderungen an die Gestaltung der graphischen Benutzerschnittstelle. Mit der Entwicklung der überwiegend graphisch gesteuerten Betriebssysteme hat sich die Anforderung von Applikationen von der reinen Zugänglichkeit der Funktionen hin zu einer möglichst angenehmen Bedienung verlagert. Gerade im Webdesign, welches aufgrund der technischen Einschränkungen schwieriger umsetzbar ist, haben unter dem Stichwort Usability zusammengefasste Aspekte einen hohen, auch wissenschaftlichen, Stellenwert. Ich konnte daher zu diesem Thema während meines Studiums mehrere Veranstaltungen besuchen[86].

2.1 Usability

Der Begriff *Usability* beschreibt *Benutzbarkeit* der komplexen Funktionsvielfalt menschlicher Technik-Artefakte und ihre Zugänglichkeit für Benutzer aller Erfahrungsstufen. Jedes Produkt muss dabei vor dem Hintergrund seiner Einsatzziele individuell gemessen werden. Im Folgenden werden einige Aspekte guter Benutzbarkeit beleuchtet.

Individualisierbarkeit

Moderne Applikationen haben weitgehende Möglichkeiten ihre Bedienoberflächen anzupassen. Techniken wie das *DOT-NET-Framework* liefern diese Funktionsvielfalt in ihren GUI-Werkzeugen von Haus aus. Diese Anpassungen der Menüstruktur oder der Shortcut-Leisten ist nicht auf Webdesign übertragbar. Dennoch können in einem gewissen Rahmen Anpassungsoptionen gegeben sein. Ein Beispiel ist die Unterstützung von Sprachen – Stichwort Internationalisierung. Dennoch sind der Individualisierbarkeit im Web nach heutigen technischen Stand Grenzen gesetzt. Denn Webseiten, die keinen Login anbieten, müssen zur Speicherung von Individualisierungen auf *Cookies* zurückgreifen, deren Verfügbarkeit nicht vorausgesetzt werden kann.

Plattformunabhängigkeit

Ein immanenter Vorteil von Webanwendungen ist ihre Unabhängigkeit von einer festen Plattform. Trotzdem gibt es darüber hinaus zu beachten, dass der Webdesigner vor der Aufgabe steht eine neue, unabhängige *Corporate Identity* für eine Webseite zu schaffen, während der Windows, Linux, Macintosh oder Java-Programmierer sich lediglich an die jeweiligen Vorgaben seiner Entwicklungsplattform halten muss. Dies erhöht die Vielfältigkeit der Möglichkeiten, erfordert aber auch mehr Aufwand schon bei kleinen Projekten.

86 Siehe Szwillus, Usability Engineering 2004/2005, [O13] oder auch Tauber, User Centered Web Design 2004.

Kognitive Unterstützung

Dynamische Oberflächenveränderungen erleichtern die Benutzerführung. Das *Highlighting* von mit der Maus überfahrenen Schaltern ist ein wichtiges Beispiel für die Unterstützung des Benutzers in seiner Hand-Augen-Koordination. Leider stehen diese Effekte nicht automatisch auf einer Webseite zur Verfügung. Lediglich Links werden durch die Veränderung des Mauszeigers verdeutlicht, allerdings nur dann, wenn sich der Zeiger tatsächlich direkt über dem Text befindet.

Weiterhin ist die Unterstützung des Benutzers bei der Behebung von Eingabefehlern wichtig. Das System sollte gutmütig menschliche Eingaben interpretieren, allerdings niemals zweideutige Angaben ohne Nachfrage übernehmen.

Technisch lassen sich diese unterstützenden Maßnahmen mit Hilfe von *JavaScript* umsetzen. Ohne die Benutzung von *JavaScript* zur notwendigen Voraussetzung zur Betrachtung einer Webseite zu machen, sind unterstützende Eingriffe ein sehr geeignetes Einsatzgebiet von dynamischem HTML.

Gestaltung und Farben

Farbgestaltung gehört nicht nur in den künstlerisch-schaffenden Bereich. Es gibt handfeste Kriterien, die auch Programmierern bekannt sein sollten. Die Beobachtung erfolgreicher Softwareprodukte motiviert die These, dass auch optische Gestaltung ein Kriterium erfolgreicher Usability ist.

Ein Beispiel ist das *Aqua* genannte Oberflächendesign des Macintosh-Betriebssystems. Die ansprechende Ansicht ist ein Merkmal des Markterfolges von Apple-Produkten. Es flossen entscheidende Elemente von *Aqua* in Produkte anderer Firmen ein, nicht zuletzt in Oberfläche von Windows Vista.

Um erfolgreich am Markt zu bestehen, scheint aber nicht die Aufwendigkeit einer Gestaltung das entscheidende Kriterium zu sein. Das Beispiel für erfolgreichen Purismus ist *Google*. In den Anfangstagen des Internet gab es viele Seiten, die der Orientierung im Netz dienten. Anbieter wie *Fireball* und *AltaVista* hatten keinen Unterschied in der Funktionsweise und keinen offensichtlich erkennbaren Qualitätsunterschied in den Ergebnissen Trotzdem war *Google* ungleich erfolgreicher. In dem Zusammenhang fällt die Schlichtheit der *Google*-Oberfläche auf. Das Webarchiv[87] liefert einen Screenshot der bis heute nahezu unveränderten *Google*-Oberfläche und dazu einen Vergleich der mit vielen Informationen befüllten Präsenz von *AltaVista* aus dem März 2000. Noch heute findet sich auf der Google-Startseite nur eine einzige Grafik. Dieses Logo liegt bei einer Dateigröße von kleinen 9 Kilobyte im GIF-Format vor. Der Effekt „mal eben“ im Netz suchen zu können wird durch die schnellen Ladezeiten hervorgehoben, was besonders im Jahr 2000 (Netzzugang per Modem) ein wichtiger Vorteil gewesen sein muss.

87 http://www.archive.org

Schönes Design

Schönheit liegt im Auge des Betrachters Daher ist eine Anpassbarkeit der Oberflächenstruktur, der Farben und die Verwendung von Skins immer eine gute Option. Es gibt einige grundlegende biologische Grundlagen der menschlichen Wahrnehmung von Farben, welche direkte Auswirkungen auf die Wahrnehmung von graphischen Benutzeroberflächen haben. Es ist daher ein Irrtum, zu meinen, Schönheit läge *nur* im Auge des Betrachters. Tatsächlich kann der überwiegenden Mehrheit der Benutzer eine Oberfläche geboten werden, welche grundsätzlich als angenehm empfunden werden wird.

Farben im Auge

Für das Erkennen von Licht sind im menschlichen Auge die auf der Netzhaut angeordneten *Stäbchen* und *Zapfen* zuständig. Die Stäbchen sind für die Unterscheidung von hell und dunkel verantwortlich und die Zapfen für das Farbensehen. Dabei übersteigt die Anzahl Stäbchen auf der Netzhaut die der Zapfen etwa um den Faktor 20. Zusätzlich sind die Zapfen wesentlich konzentrierter im Sehzentrum der Netzhaut angeordnet, während die Stäbchen gleichverteilter liegen. zeigt diese graphisch. Weiterhin untergliedern sich die Zapfen in drei Subtypen, welche Rot, Grün und Blau unterscheiden können[88]. Daraus ergeben sich folgende Konsequenzen für die Empfindung von Farben einer Benutzeroberfläche auf dem Monitor.

1. Die Tatsache mehr Stäbchen als Zapfen zu haben führt dazu, Konstraste unterschiedlicher Helligkeiten deutlicher wahrzunehmen als Kontraste unterschiedlicher Farben. Zudem reagieren die Stäbchen schon bei geringeren Reizen als die Zapfen[89]. Konsequenz: Der Mensch wird grundsätzlich dunkle Schriftfarben mit hellen Hintergründen oder umgekehrt als angenehm empfinden. Schriften, die sich nur durch den Farbton vom Hintergrund abheben, lassen uns den Text schlechter erkennen.

2. Die drei Subtypen der Zapfen eröffnen Möglichkeiten flächige Elemente voneinander abzugrenzen. Dies ist durch die Verwendung von Farben der gleichen Helligkeit in einem anderen Farbton möglich, ohne einen starken Konstrast zu erzeugen. Komplementärfarben haben maximal auseinanderliegende Farbbereiche bei gleicher Helligkeit.

3. Aus der punktuellen Konzentration der Zapfen im Sehzentrum folgt die schlechte Eignung von leuchtenden Farben für große Flächen und Hintergründe. Das Auge kann diese Fläche nicht gut fokussieren und das Ergebnis ist – subjektiv empfundene – Unruhe und Hektik im Bild. Umgekehrt kann leuchtende Farbe sehr gut Einzelheiten hervorheben, beispielsweise durch Icons.

88 Das Modell ist vereinfachend, denn in Wirklichkeit können die Zapfen nicht Rot, Grün und Blau erkennen, sondern reagieren auf unterschiedliche Wellenlängen elektromagnetischer Strahlung. Diese Reaktion interpretiert das menschliche Gehirn als eine der drei Grundfarben und setzt daraus das gesamte Bild zusammen.

89 Dies ist die Ursache dafür, bei schlechten Lichtverhältnissen keine Farben mehr erkennen zu können, wohl aber Helligkeitsunterschiede.

4. Die drei Zapfentypen sind unterschiedlich auf der Netzhaut verteilt. Die Rezeptoren für Blau liegen gleichmäßiger und die Rot-Rezeptoren sind punktuell stärker im Sehzentrum konzentriert. Konsequenz: Rot sollte für punktuelle Elemente, wie Warnhinweise verwendet werden, Blau kann dagegen auch als Hintergrundfarbe dienen.

Psychologische Einflüsse

Neben der physiologischen müssen auch psychologische Einflüsse berücksichtigt werden. Das Gehirn übernimmt die Signale des Auges nicht einfach nur, sondern interpretiert die Reize. Dies ist ein notwendiger Filter zur Reduzierung der sonst unfassbaren Informationsmengen und geschieht unterbewußt anhand der individuellen Erfahrungen jedes Menschen. Einige bekannte optische Täuschungen basieren darauf. zeigt für die Farbwahrnehmung interessante Beispiele. Links wird der Einfluss unserer Erfahrung auf unsere Wahrnehmung unterstrichen. Kinder sehen im Bild nichts außer Delphine. Das rechte Bild demonstriert eine gewollte Farbverfälschung durch das Gehirn. Feld B hat die gleiche Hintergrundfarbe wie das Feld A und dennoch empfinden wir es als dunkler. Das Gehirn bezieht in diesem Fall unsere Kenntnis des Schattens mit ein. Es lassen daraus sich einige Schlussfolgerungen für die Gestaltung ziehen.

1. Prinzipiell wird ein Text „schwarz auf weiss" als bekannt und angenehm empfunden, weil wir es aus täglichen Geschäftsprozessen so gewohnt sind.

2. In der Natur kommen keine planen Flächen vor. Über die eigentliche Struktur hinaus sorgt der Lichteinfall auch bei Artefakten für eine graduelle Helligkeitsverteilung. Daher kommt dem Menschen eine große, auf dem Monitor uniform gleich leuchtende Fläche als unnatürlich und künstlich vor. Die Verwendung von Schattierung wird als angenehm empfunden. Dieser Eindruck ist überwältigend wichtig und wird von jeder modernen Computerapplikation intensiv benutzt. Unter Windows gibt es beispielsweise fast keine Fläche mehr, die nicht mit einen leichten Gradienten texturiert wäre, dies ist ebenso ein Erfolgsgeheimnis von *Aqua*.

3. Bei natürlichem Lichteinfall wirft grundsätzlich jeder Gegenstand Schatten. Die unterschiedliche Helligkeit einer Farbe beim Übergang zum Schatten wird vom Gehirn ausgeglichen. Schatten kann die Oberfläche natürlicher wirken lassen. Ähnliches gilt für Glanz und Spiegeleffekte.

4. Bildliche Darstellung erzeugen im Gehirn eine unmittelbare Assoziation. Texte hingegen sind nur die Kodierungen von Daten. Diese müssen erst vom Verstand übersetzt und verstanden werden. Daher fördern Piktogramme, Icons und andere bildliche Darstellungen die Eingänglichkeit.

5. Auch das Webdesign selber hat schon einen Einfluss auf unsere Informationsbewertung genommen. Ein unterstrichener Text erzeugt die Assoziation weitergehender Informationsmöglichkeiten über einen Link. Strukturinformationen werden intuitiv links oder in einer Zeile oberhalb der Hauptinformation gesucht.

3. Projektmanagement

Die Standish Group [O10] gibt seit 1994 in regelmäßigen Aktualisierungen den Chaos-Report[90] heraus. Dieser untersucht Erfolgs- und Misserfolgsfaktoren von IT-Projekten. Die Zahlenbasis der Studie beruht auf der wissenschaftlichen Untersuchung von über 40.000 Einzelprojekten[91]. Demnach scheitern noch heute knapp ein Fünftel aller Softwareprojekte. Und auch wenn sich der prozentuale Anteil erfolgreicher Projekte seit 1994 mehr als verdoppeln konnte, verbleiben statistisch noch 46% aller Softwareprojekte in einem nur teilweise erfolgreichen Status. Sie verbrauchen mehr Budget oder mehr Zeit oder werden nicht in der ursprünglich vorgesehenen Funktionsvielfalt fertiggestellt. Die große Zahl der Softwareprojekte vom „Typ 2“ (nur teilweise erfolgreich) zeigt, dass die Ursache für das Scheitern nicht in der technischen Unlösbarkeit der Ziele, sondern im Prozessmanagement liegt.

Konkret werden in den der Chaos-Studie zugrunde liegenden Befragungen als Ursache für das Scheitern von IT-Projekten fehlende oder unklare Anforderungen und häufige Anforderungsänderungen genannt[92]. Ein Grund für den deutlich größeren Anteil erfolgreicher Projekte der Studie 2007 seien zunehmend zum Einsatz kommende iterative Arbeitsweisen[93]. Daran lassen sich zwei Erkenntnisse ablesen:

1. Ein Softwareentwicklungsprozess ist mehr als die Programmierung selbst. Zur Bereitstellung einer wie auch immer ausführbaren Softwarekomponente sind weitere Disziplinen gefordert, beispielsweise Anforderungsmanagement, Architekturüberlegungen, Konfigurationsmanagement und Qualitätssicherung.

2. Nach dem aktuellen Stand der Technik lässt sich kein Softwareprojekt von Grund auf planen und Anhand der Planung gradlinig ausführen, vielmehr muss von Anfang die Möglichkeit bestehen, auch grundlegende Annahmen zu verändern.

3.1 Vorgehensmodell

Diesen Erkenntnissen versuchen Vorgehensmodelle Rechnung zu tragen. Es existieren unterschiedliche Ansätze den Softwareentwicklungsprozess zu strukturieren, aber leider wenig vergleichende Literatur, gerade im

90 Der erste Chaos-Report von 1994 ist auf der Webseite der Standish Group nach Registrierung kostenfrei einsehbar. Der aktuellste Chaos-Report 2007 wird von der Standish Group für eine Summe von 1.000 Dollar angeboten. Quelle der Statistik von 2007 ist ein Bericht der Zeitschrift Computerwoche vom 12. März 2007.

91 Nach Angaben der Standish Group.

92 Siehe Chaos Report 1994, Success/Failure Profiles, dort nennen 13,1% der Befragten als „Project Impaired Factors“ unvollständige Anforderungen und knapp 9% sich verändernde Anforderungen und Spezifikationen.

93 Jim Johnson, Chairman der Standish Group in Computerwoche vom März 2007.

wissenschaftlichen Bereich. Ein Grund dafür mag in der Tatsache liegen, dass Vorgehensmodelle aus der Praxis großer Projekte entstehen und damit der marktwirtschaftlich getrieben sind[94]. Zum anderen unterliegt der Markt schnellen Veränderungen[95].

Die Modelle unterteilen den Projektablauf zunächst in mehrere, konsekutiv zu durchlaufende Phasen. Durch die zeitliche und inhaltliche Begrenzung der verschiedenen Phasen wird die Projektdurchführung überschaubar und damit in ihrer Gesamtkomplexität verringert. Als typische Vertreter dafür existieren insbesondere im deutschen Projektgeschäft das *V-Modell XT*, der *Rational Unified Process (RUP)* und das *Extreme Programming (XP)*[96].

Der RUP besitzt ein Alleinstellungsmerkmal im Hinblick auf die zu erstellenden Dokumente, Grafiken oder Quelltexte – Artefakte in der RUP-Nomenklatur – zu definierten Meilensteinen. Durch die enge Anlehnung an die Unified Modeling Language (UML) werden diese Artefakte detailliert beschrieben und entsprechende Vorlagen vom RUP bereitgehalten[97]. Die Art der Artefakte unterscheiden sich dabei jeweils nach den vom RUP definierten Disziplinen, beispielsweise einem Anforderungsmanagement oder der Implementierung.

Das V-Modell XT definiert in ähnlicher Weise Projektphasen mit zugehörigen Meilensteinen, ohne allerdings die Art der zu erbringenden Ergebnisse bis herunter auf die Dokumentenebene vorzugeben. Durch seine flexible Bausteinphilosophie ist das Modell jedoch einfacher für IT-Projekte verwendbar, die nicht ausschließlich oder gar keine Software-, sondern Hardwareerstellung zum Ziel haben[98].

Allen Vorgehensmodellen ist gemein, dass sie nicht starr – Out-of-the-Box – verwendet werden können und sollen, sondern vielmehr an die konkrete Projekt-, Unternehmens- und Personalsituation anzupassen sind[99]. Im V-Modell XT sind für diesen Zweck die genannten Bausteine vorgesehen, vom RUP existieren verschiedene Varianten für kleine, mittlere und große Projekte oder spezielle Projekte, wie diejenigen, die ein bestehendes Produkt weiterentwickeln, statt etwas komplett Neues zu schaffen.

Es war aber gerade der umfassende Organisationsansatz insbesondere des Rational Unified Process, welcher von vielen Softwareentwickern als „zu schwerfällig“ empfunden worden ist. Als Gegenstück entstanden Mitte der Neunzigerjahre die ersten agilen Modelle, die versuchen den

94 Ein Werk der Gegenwart zum Thema Software-Vorgehensmodelle ist Bunse/von Knethen, Vorgehensmodell kompakt aus der Serie IT Kompakt.
95 Eines der wenigen Bücher der Bibliothek der Universität Paderborn stammt aus dem Jahr 1992 und kennt kein RUP und XP nicht und das V-Model nur in seiner Urform von 1979. Siehe Hesse/Merbeth/Fröhlich, Handbuch der Informatik – Software-Entwicklung.
96 Siehe Hoffmann, Software-Qualität, S. 499ff.
97 Siehe Rational Unified Process, Best Practices for Software Development Teams, Rational Software Whitepaper, TP026B, Rev 11/01.
98 Siehe Hoffmann, Software-Qualität, S.499f.
99 Aus dem Englischen stammend hat sich hierfür der Anglizismus „tailor“ in Umfeld der Informationstechnologie eingebürgert – allerdings bisher ohne eine Verbreitung in den allgemeinen Sprachgebrauch.

Softwareentwicklungsprozess wieder schlanker zu gestalten ohne an Effektivität zu verlieren. Der bekannteste Vertreter dieser Philosophie ist das Extreme Programming. XP führt einige innovative Konzepte ein, wie das Pair-Programming. Auch die intensive Nutzung von Komponententests gehen mit der Entstehung von agilen Modellen einher.

Der Erfolg des oben genannte Einsatzes von automatisierten Komponententests[100] führte zur Formulierung des *Test-Driven-Development (TDD)* als eigenständige Vorgehenstechnik. Dieses „Modell“ sieht zunächst die Erstellung von Tests für eine zu erstellende Softwarekomponente vor. Die folgende Implementierung kann dann jederzeit – und insbesondere direkt nach ihrer ersten Erstellung – anhand der definierten Tests evaluiert werden. Die Hervorhebung eines solchen einzelnen Aspektes (in diesem Fall des Aspektes „Verifikation“) wird häufig ebenfalls als Vorgehensmodell bezeichnet, Beispiele dafür sind neben Test-Driven-Development auch *Model-Driven-Development*[101], *Prototyping*[102] oder *Feature-Driven-Development*[103]. Der Umfang dieser Methoden und die Charakterisierung als Vorgehensmodell oder lediglich als einzelne Technik sind fließend. Auch XP ist letztlich eine Zusammenstellung von Techniken, die sich als Best Practices aus der Praxis ergeben haben. Diese Summe der Techniken verdichtet sich erst zu einem kompletten Vorgehensmodell.

Es gibt Stimmen[104], welche die Schaffung eines Vorgehensmodells zur ingenieurmäßigen Herstellung von Software grundsätzlich ablehnen. Im Kern geht es um die Frage, ob es sich bei der Erstellung von Software um einen künstlerischen Akt oder schlicht um einen herstellenden Vorgang handelt. Insbesondere die Beobachtung des Open-Source-Marktes zeigt an den Beispielen OpenOffice und Mozilla oder auch dem Apache-Projekt[105], dass einzelne Produkte ohne die Kreativität und Ausdauer von Entwicklern, die sich als Freischaffende

---

100 Häufig Unit-Test genannt. Der Text folgt der Nomenklatur des International Software Testing Qualifications Board (ISTQB). Unter Unit-Testing wird vielfach lediglich die Verwendung eines Frameworks wie jUnit oder vergleichbare Derivate für andere Programmiersprachen verstanden.

101 Model-Driven-Development (MDD) betont die Voraussetzung von Modellierung für die Erstellung von Software. Als Modellierungssprache kommen UML oder domänenspezifische Sprachen (DSL) zum Einsatz. Siehe dazu beispielsweise Stahl/Völter/Efftinge, Modellgetriebene Softwareentwicklung. Techniken, Engineering, Management.

102 Ziel von Prototyping ist die schnelle Erstellung einer lauffähigen Beispielanwendung für schnelles Feedback an Auftraggeber und Auftragsnehmer. Prototypen können weiterentwickelt (evolutionäres Prototyping) oder komplett verworfen werden. Siehe dazu beispielsweise Kruchten, The Rational Unified Process, S. 185ff.

103 Feature-Driven-Development (FDD) bezeichnet im Kern die jeweilige Konzentration auf eine einzelne Funktion, die eine Software dem Nutzer zur Verfügung stellt. Diese Funktion wird quer durch alle Softwareschichten zeitlich zusammenhängend implementiert. Dies steht im Gegensatz zur Technik, schichtenweise zu arbeiten und zunächst eine komplette Datenbankabstraktionsschicht zu implementieren. Siehe beispielsweise Palmer/Felsing, A Practical Guide to the Feature-Driven Development. Siehe auch Webseite des Feature-Driven Development [O11].

104 Siehe dazu Hoffmann, Software-Qualität, S. 540ff.

105 Dieser Aspekt wird von Ralf S. Engelschall in „Open-Source: Freiheitskampf zwischen Kunst und Kommerz“, ObjectSpectrum 02/2009, S. 44ff am Beispiel Apache-Erweiterungsmodels „mod_ssl“ aufgezeigt. Die Firma Google trägt der Erkenntnis Rechnung, indem Sie jedem Mitarbeiter erlaubt, 20% seiner Arbeitszeit in eigene Projekte zu investieren.

verstehen, überhaupt nicht denkbar gewesen wären.

3.2 Zwischenergebnis

In jedem Fall bleibt festzuhalten, dass professionelle Softwareentwicklung in großen Projekten und unter Beteiligung von mehreren Teammitgliedern ohne eine Aufteilung in Disziplinen wie Anforderungsmanagement oder Konfigurationsmanagement nicht mehr denkbar wäre. Um diese Tätigkeiten verzahnen zu können sind Vorgehensmodelle notwendig. Der Zuschnitt der Vorgehensmodelle auf die konkrete Projektsituation führt nicht selten zur Mischung verschiedener Ansätze. Alle Vorgehensmodelle stammen direkt aus der Praxis. Es fehlt bisher ein externer wissenschaftlicher Zugang zu der Thematik. Insofern sind umfassende Darstellungen rar, insbesondere eine Bewertung oder Empfehlung für ein Modell ist zur Zeit schwierig. Der Rational Unified Process besitzt ein Alleinstellungsmerkmal in der Definition seiner Artefakte. Dies fördert seine Eignung für die Dokumentation eines Projektfortschritts. Im Folgenden werden daher andere Ansätze nicht weiter betrachtet.

3.3 Rational-Unified-Process

Der Rational Unified Process (RUP) ist ein Vorgehensmodell der Softwareentwicklung,. Es wird von der Firma IBM entwickelt und vermarktet. RUP steht seinen Lizenznehmern in Form eines verlinkten Webdokuments zur Verfügung, es wird wie Software durch regelmäßige Aktualisierungen seitens des Herstellers gepflegt. Die Webanwendung hält Handbücher, Dokumentenvorlagen und Anleitungen zur Werkzeugunterstützung bereit. IBM bietet auf den RUP zugeschnittene Werkzeuge an, unter anderem das bekannte Rational Rose zur Modellierung oder die Versionsverwaltung Rational Clearcase. In seiner Urform wurde der RUP 1996 von Philippe Kruchten vorgestellt[106]. Insgesamt ist der RUP und seine Nomenklatur in Buchform veröffentlicht[107] und damit frei verfügbar, die Webanwendung, Vorlagen und Werkzeuge sind dagegen lizenzpflichtig.

Der RUP vereint Best Practices zu einem Produkt. Sein Vorgehen ist grundsätzlich iterativ organisiert und berücksichtigt Anforderungsänderungen. Die Softwareintegration erfolgt kontinuierlich mit dem Ziel einer schnellen und immer verfügbaren Einsicht in den Projektstatus für das Projektteam und weitere Beteiligte. Das Anforderungsmanagement nimmt eine Schlüsselrolle im Prozess ein. Anforderungen werden systematisch erfasst und durch Reviews und Tests permanent überprüft. Zur Bewältigung von Anforderungsänderungen sind Change-Management-Prozesse und das Configuration-Management vorgesehen. In allen Disziplinen des RUP nimmt modellgetriebene Entwicklung eine Schlüsselrolle ein. Die Grundlage dafür ist die Unified Modeling Language

106 Siehe Kruchten, The Rational Unified Process – An Introduction.

107 Philippe Kruchten beschreibt in „The Rational Unified Process" nach eigener Aussage nur „eine kleine Untermenge" des RUP, gibt aber insgesamt eine komplette Beschreibung der iterativen Ansätze, aller Rollen, Aktivitäten und Arbeitsabläufe. Weitere Informationen lassen sich aber in vielen weiteren Veröffentlichungen finden, beispielhaft genannt sei Jacobson/Booch/Rumbaugh, The Unified Software Development Process.

(UML). Der RUP ist damit implizit objektorientiert. Das Modell betont einen methodischen und systematischen Ansatz zur Modellierung komponentenbasierter Softwarearchitekturen[108].

Um allen Umgebungen gerecht zu werden, soll der RUP an die jeweilige Projektsituation angepasst werden, dies wird als Tailoring bezeichnet. Dazu existieren von Haus aus bereits unterschiedliche Varianten des RUP[109], beispielsweise Ausprägungen für kleine oder große Projekte oder für Projekte mit einer speziellen Zielsetzung, wie Wartungs- oder COTS-Projekte[110]. RUP für Wartungsprojekte beschreibt detailliert, wie eine Menge von Bugs und Change-Requests in ein inkrementelles Release eingebracht werden und RUP für COTS-Projekte geht auf Abwägungen zwischen Anforderungen, Risiken und der Marktsituation ein. Im Allgemeinen verspricht jedoch eine zusätzliche Anpassung des Prozessmodells Vorteile. Die Idee dieses sogenannten Tailorings ist so immanent mit dem RUP verbunden, dass IBM mit dem Rational Method Composer ein eigenes Werkzeug für diesen Zweck anbietet.

3.4 Dimensionen

Die Prozessstruktur des Rational Unified Process hat zwei Dimensionen, ist eine Übersicht am Beispiel des klassischen RUP.

- Die horizontale Achse zeigt die zeitliche Organisation eines Projektes. Die Iterationen sind in vier Phasen aufgeteilt, die jeweils mit eigenen Meilensteinen schließen. Diese Dimension zeigt die dynamische Struktur des RUP.
- Die vertikale Achse zeigt die inhaltliche Organisation eines Projektes. Die Arbeitsabläufe sind in neun Disziplinen unterteilt. Diese Dimension repräsentiert den statischen Aspekt des Prozesses.

Prinzipiell existiert nach der Philosopie des Unified-Process jede Disziplin in jeder Iteration des Projektes. Die Disziplinen arbeiten jeweils parallel oder sequentiell und verdichten ihre Artefakte in jeder Iteration weiter bis zum Endresultat. Das impliziert insbesondere, dass die Disziplin Test nicht erst nach Abschluss der Implementierung startet, sondern von Anfang an die Qualität des Produktes sichert[111]. Dennoch sind nicht alle Disziplinen zu allen Zeiten gleich umfangreich gefordert. Anforderungen beispielsweise werden zu Anfang des Projektes vor allem in der Inception Phase aufgenommen, aber eben nicht ausschließlich dort. Dies stellen die Kurven der Übersichtstafel in dar. Der exakte Verlauf der Kurve ist projektspezifisch. Die Grafiken sollten als statistischer

---

108 Siehe dazu Kruchten, The Rational Unified Process, S. 17ff.

109 Siehe dazu Kroll/MacIsaac, Agility and Discipline Made Easy – Practices from OpenUP and RUP, S 18 und besonders Anhang B.

110 Die Abkürzung COTS steht für Commercial-Of-The-Shelf und bezeichnet damit Software von der Stange, die also ohne indivuelle Softwareanpassungen verkauft werden. Ein Beispiel ist Standard-PC-Software wie ein Office-Paket.

111 Diese Einschätzung hat sich mittlerweile durchgesetzt und wird im Syllabus des International Software Testing Qualifications Board (ISTQB) so gefordert. Siehe dazu auch Spillner/Linz, Basiswissen Softwaretest, 3. Auflage, S 35f.

Mittelwert verstanden werden.

3.5 Phasen

Jedes Projekt nach dem Rational Unified Process ist zwingend in vier Phasen unterteilt. Die Länge der einzelnen Phasen hängt stark vom *Tailoring* des Prozesses und somit vom konkreten Projekt ab. Die Phasen sind:

1. Inception Phase
2. Elaboration Phase
3. Construction Phase
4. Transition Phase[112]

Am Ende jeder Phase steht ein Meilenstein, an dem festgelegte Kriterien erfüllt sein müssen, damit das Projekt in die nächste Phase überführt werden kann. Die Transition-Phase schließt mit einem fertigen Release des Softwareproduktes ab. Nicht selten ist dieses Release Eingangsgröße für einen weiteren sogenannten Zyklus, der erneut alle vier Phasen durchläuft und mit einem weiteren Release endet.

Die folgende Darstellung der Phasen und der Disziplinen folgt dem Buch von Philippe Kruchten [P02].

Inception-Phase

In der *Inception-Phase* wird Umfang und Abgrenzung des Projektes festgelegt. Ziel ist eine Definition darüber anzugeben, was die Software leisten soll, und mit gleichem Stellenwert, was sie nicht leisten soll. Darauf aufbauend werden Abnahmekriterien definiert, eine sehr frühe Einbindung des Auftragsgebers – dem Kunden – ist essentiell. Das Ergebnis ist das Visionsdokument. Kritische Systemkomponenten und wesentliche Szenarios werden schon in dieser Phase in abtrakten *Use-Cases* modelliert. Mindestens eine mögliche Realisierungsachritektur wird herausgestellt und eventuell vorgeführt, dazu wird das *Prototyping*[113] eingesetzt. Ein zentrales Element des RUP ist die Abschätzung von Risiken[114] in dieser frühen Projektphase.

Am Ende der *Inception-Phase* steht der *Lifecycle Objectives Milestone (LCO)*. An diesem Meilenstein muss eine Übereinstimmung aller Beteiligten[115] über den Umfang, die Kosten und den Zeitplan herrschen. Weiterhin muss ein gemeinsames Verständnis der Anforderungen existieren. Wichtig ist dabei Prioritäten und Risiken realistisch einzuschätzen. Der RUP fordert einen ehrlichen Umgang unter Umgehung betriebswirtschaftlicher Schönungen. Der Prototyp muss eine allgemeine Akzeptanz im Entwicklungsteam, beim Management und beim

112 Die englischen Bezeichnungen werden als definierte Nomenklatur des RUP beibehalten.
113 Für weitere Details siehe Abschnitt 3.6 Prototypen auf Seite 56.
114 Siehe Kroll/MacIsaac, Agility And Discipline Made Easy, S. 29ff. Zur Begegnung von Risiken siehe Seite 53, Stichwort Risikomanagement.
115 In der Nomenklatur des RUP: Stakeholder.

Auftragsgeber finden. Am Ende der *Inception-Phase* werden die für diese Phase veranschlagten Kosten und der Zeitplan mit den Planungen abgeglichen.

Sollte in einigen dieser Punkte keine Übereinkunft gefunden werden, so ist eine weitere Iteration notwendig. Es darf nicht mit der nächsten Phase begonnen werden. Ohne Übereinkunft wird das Projekt abgebrochen.

Elaboration-Phase

In der Entwurfsphase wird ein tiefgehender Einblick in das System gewonnen. Die Analyse des Problemumfeldes schließt auch nichtfunktionale Anforderungen mit ein und mündet in einer Architekturgrundlage. Die größten Projektrisiken werden mit den Methoden des Risikomanagements ausgeschaltet. Der Projektplan wird entwickelt.

Die Elaboration Phase ist eine kritische Phase, da wichtige Entscheidungen getroffen werden, die aufgrund ihres grundlegenden Charakters auch in einem iterativen Projekt weitreichende Folgen haben. Häufig bedeutet diese Phase den Übergang von einem flexiblen in ein schwerfälligeres Projekt mit höheren Kosten. Eine Ursache dafür ist der einsetzende höhere Personalaufwand und der größere Bedarf an Resourcen.

Ziel der Phase sind die Definition und die Validierung der Architektur und allgemein die Schaffung einer Grundlage[116] für die anschließende *Construction-Phase*. Die Realisierbarkeit des Projektes mit der vorgeschlagenen Architektur im gesetzten Kosten- und Zeitrahmen wird dargelegt – durch Prototyping und Expertenschätzung. Dazu werden einzelne Architekturkomponenten integriert und eine Auswahl getroffen, welche der Komponenten zu entwickeln, zu kaufen oder wiederzuverwenden sind[117]. Die ursprünglichen Szenarien aus der *Inception-Phase* sind einander gegenüberzustellen. Falls notwendig, kann hier bereits ein Redesign erfolgen. Weiterhin wird das Visionsdokument weiterentwickelt und ein solides Verständnis der Use-Cases erzeugt. Zuletzt werden in dieser Phase die Infrastruktur, also die Entwicklungsumgebung und alle relevanten Werkzeuge abschließend eingerichtet.

Das Ergebnis der *Elaboration-Phase* ist ein Use-Case-Modell mit einer geschätzten Vollständigkeit von 80% sowie ergänzenden, insbesondere nichtfunktionalen Anforderungen. Ein Prototyp befindet sich in einem ausführbaren Status. Es existiert eine überarbeitete Risikoliste und ein Projektplan, der einen Entwicklungsplan samt aller Iterationen und

---

116 Philippe Kruchten verwendet den Begriff Baseline.

117 Dies ist die Übernahme des Make-Or-Buy-Prinzips aus der Betriebswirtschaftlehre in die Informationstechnologie. Solche Abwägungen erhalten aufgrund des großen Marktes frei verfügbarer Komponenten eine besondere Bedeutung. Zu beachten ist, dass der Griff zum freien Bibliothek nicht immer der einfachste ist, beispielsweise beim Einsatz von Software in kontrollierten Märkten, wie dem Bankensektor oder dem IVD-Markt. Eine weitere Dimension erhält der Themenkomplex aufgrund der Möglichkeit Eigenentwicklungen wiederzuwenden und zu dem Zweck a priori mit generischeren Schnittstellen auszurüsten, als dies prinzipiell notwendig wäre. Siehe zum Make-Or-Buy in der IT eine Sonderausgabe der „HMD – Praxis der Wirtschaftsinformatik“ [M01] und zum Einsatz von Software in der In-Vitro-Diagnostic die Webseiten der amerikanischen Food And Drug Administration [O12].

Evaluierungskriterien enthält. In dieser Phase kann bereits mit einem vorläufigen Benutzerhandbuch begonnen werden.

Am Ende der *Elaboration-Phase* steht der *Lifecycle-Architecture-Milestone (LCA)* als zweiter Projektmeilenstein. Dieser überprüft die Stabilität des Produktes, die Vision zum Produkt und die Architektur. Prototypen müssen zeigen, dass die wesentlichen Risiken erkannt und aufgelöst wurden und es stellt sich die Frage, ob die Annahmen für die folgende *Construction-Phase* glaubhaft untermauert werden können. Eine Übereinstimmung aller Beteiligten muss darin gewährleistet sein. Es werden erneut die bisherigen Kosten mit den kalkulierten Kosten verglichen. Sollte das Projekt an diesem Meilenstein scheitern, so wird es abgebrochen oder wirklich sehr gründlich überdacht.

Construction-Phase

In der *Construction-Phase* verschiebt sich der Schwerpunkt des Projektes von der Entwicklung eher intellektueller Konstrukte zur Entwicklung einsatzfähiger Produkte. Die Herausforderung liegt im Resource-Management, im Controlling, der Kostenoptimierung, in Zeitersparnis und Qualitätssteigerung. In großen Projekten ist eine hohe Parallelität der Entwicklung möglich, wenn in den vorhergehenden Phasen eine ausreichend verteilte Architektur konzipiert wurde.

Die Formulierung der Ziele ist ungleich einfacher als deren Umsetzung. Das Ziel ist die Erstellung von Test-Releases und produktiven Releases in möglichst kurzer Zeit, einem möglichst vollständigen Funktionsumfang, unter Erreichung der geforderten Qualität und innerhalb des gesteckten Budget. Möglich wird die optimale Arbeitsweise in allen vier Dimensionen durch die Tätigkeiten der beiden ersten Phasen. Die Aktivitäten dieser Phase sind somit die Implementierung und Integration, sowie das gewissenhafte Testen auf Basis der Anforderungen. Das Ergebnis dieser Phase ist das Softwareprodukt, das Benutzerhandbuch und eine Beschreibung des aktuellen Release.

Am Ende der *Construction-Phase* steht der Meilenstein *Initial-Operational-Capability (IOC)*. Zu diesem Zeitpunkt muss entschieden werden, ob die Software in Produktion gehen kann. Sie muss dazu eine ausreichende Stabilität besitzen und alle Beteiligten – insbesondere der Auftragsgeber – müssen darin übereinstimmen, dass die Software bereit für den Übergang ist. Die aufgelaufenen Kosten werden in Relation zu den Planungen gestellt. Sollte die Software an diesem Meilenstein scheitern, so wird die *Construction-Phase* um ein Release verlängert.

Transition-Phase

In der *Transition-Phase* findet die Integration des Release in die Produktivumgebung statt. Es wird nur noch Fehlerbehebung und im Ausnahmefall das Ausprogrammieren zuvor zurückgestellter Funktionen vorgenommen.

Das Ziel ist eine endgültige Produktreife. Dazu gehört es, den Endbenutzer in die Lage zu versetzen, das System unabhängig vom Hersteller nutzen zu können. Es muss bei allen Beteiligten Einigkeit herrschen, dass die Anforderungen an das Produkt erfüllt sind.

Konkrete Aktivitäten hängen von der Art der Software ab. In jedem Fall finden dokumentierte Abnahmetests statt, um das System im Vergleich zu den Erwartungen des Benutzers zu evaluieren. Denkbar wäre ein paralleler Einsatz zu einem Altsystem oder die Anbindung von externen Datenbanken. Im Fall von Standardsoftware findet eine kommerzielle Fertigstellung statt, Verpackung, Marketing und die Übergabe an den Vertrieb. Bei Auftragssoftware sind die Benutzer und Administratoren zu schulen.

Das Ergebnis dieser Phase ist ein fertiges Produkt samt Schulungsunterlagen oder Verkaufsmaterial. Die *Transition-Phase* kann je nach Produkttyp sehr unterschiedlich sein.

Am Ende dieser Phase steht der Meilenstein *Product-Release (PR)*. Zu dieser Zeit wird entschieden, ob das Produkt vollständig ist. Die Kosten werden abschließend gegen die Budgetplanungen gerechnet. Dieser Meilenstein kann bereits mit dem Ende der *Inception-Phase* des nächsten Produktionszyklus zusammenfallen.

3.6 Disziplinen

Der Rational-Unified-Process beschreibt die Antworten auf die Fragen nach dem *Wer*, *Was*, *Wann* und *Wie*. Die Tätigkeiten einer Disziplin werden als Konsequenz durch vier elementare Modellierungselemente beschrieben:

- Eine Liste der *Worker* definiert das *Wer*, also die an einer Aufgabe beteiligten Rollen. Beispiele für Rollen sind ein Designer oder auch ein technischer Architekt.
- Die Rollen werden mit *Activities* verknüpft, welche das Wie definieren. Zu diesen Aktivitäten gehören die Planung einer Iteration, ein Design-Review oder die Ausführung von Performance-Tests.
- Der *RUP* bezeichnet jede Arten von Produktergebniss als Artefakt, zum Beispiel Quelltexte, kompilierte Programme, Textdokumente oder Modelle. Die Ensemble der Artefakte stellen das *Was* dar.
- Die Workflows grenzen die Frage nach dem *Wann* ein. Ein Workflow ist eine Sequenz von Aktivitäten. Jeder Workflow lässt sich einer Disziplin zuordnet.

Project-Management

Die Disziplin *Project-Management* bietet ein Framework zum erfolgreichen Managen der Entwicklung Software-lastiger Systeme. Dieser Workflow des Rational Unified Process deckt dabei bewusst nicht alle Aspekte eines erfolgreichen (Gesamt-) Projektmanagements ab; Vertragsverhandlungen, Mitarbeiterführung und -anwerbung und dergleichen bleiben im Prozessmodell unberücksichtigt.

Zwei der Schlüsselkonzepte zum effektiven Planen mit Hilfe des RUP sind die Konzepte des Risikomanagements und die Beachtung von Metriken.

Risikomanagement

Auch gute Planungsprozesse erfassen zunächst nur die bekannten Aspekte eines Projektes. Das Risikomanagement behandelt die Aspekte eines Projektes, die bisher unbekannt sind. Risiken sind Faktoren, die den Erfolg des Projektes gefährden oder verhindern können. Risiken treten mit einer gewissen Wahrscheinlichkeit ein und Risiken haben unterschiedliche Schwere, ihr Eintritt gefährdet das Projekt unterschiedlich stark. Auf einige der Risiken hat das Projekt direkten Einfluss, andere liegen außerhalb des Einflussbereiches der Projektleitung. Die Erkenntnis eines Risikos ist im Sinne des RUP nicht ausreichend, vielmehr wird jedes Risiko adressiert und behandelt. Hierfür sind vier Strategien zu nennen:

1. **Risk-Avoiding**: Das Risiko wird vermieden, also ausgeschaltet.
2. **Risk-Transfering**: Das Risiko wird auf einen anderen Träger geschoben.
3. **Risk-Reducing**: Die Eintrittswahrscheinlich des Risikos wird aktiv reduziert.
4. **Risk-Accepting**: Die Existienz des Risikos wird akzeptiert. Ein alternativer Plan für den Fall des Eintretens wird erarbeitet.

Eine Methapher gibt ein klares Verständnis der Begegnungsstrategien. Die Vorstellung ist ein Wohnhaus direkt neben einem großen, alten und innerlich verfaulten Baum. Das Risiko ist das Kippen des Baumes auf das Wohnhaus und dessen Zerstörung mit einer gewissen wetterabhängigen Wahrscheinlichkeit.

1. **Risk-Avoiding**: Der Baum wird von Fachleuten kontrolliert gefällt. Im Ergebnis existiert das Risiko nicht mehr
2. **Risk-Transfering**: Der klassische Fall zum transferieren eines Risikos ist der Abschluss einer Versicherung, in diesem Fall einer Gebäudeversicherung gegen Naturschäden.
3. **Risk-Reducing**: Um die Eintrittswahrscheinlich zu Reduzieren wird der Baum durch ein Netz, einen Betonkern oder ähnliches gegen Kippen gesichert.
4. **Risk-Accepting**: Das Risiko wird als Vorhanden akzeptiert und vor dem Hintergrund der Wetterlage als nicht zu kritisch gesehen, ohne die Augen vor der Gefahr zu verschließen. Eine alternative Wohnmöglichkeit wird gesucht und bereitgehalten.

Das Beispiel zeigt, dass der intuitive Ansatz, ein Risiko grundsätzlich vermeiden zu wollen, meist gar nicht möglich ist, oder dramatische Auswirkungen hat. Weiterhin bedeutet das Akzeptieren des Risikos nicht, einfach nichts zu tun, sondern vielmehr einen Alternativplan bereit zu halten. Sinnvoller ist daher häufig die Risikominimierung oder – wenn möglich – die Verlagerung des Risikos, beispielsweise auf den Auftragsgeber.

Metriken

Mit Metriken soll der Versuch unternommen werden, messbar den Erfolg oder den aktuellen Status eines Softwareprojektes beurteilen zu können. Da die Erfassung von Metriken Kosten verursacht, können nicht einfach alle Aspekte gemessen werden, die überhaupt messbar sind und nicht alle Aspekte, die interessieren, sind überhaupt messbar. Der Einsatz von Metriken muss vernünftig abgewogen werden. Metriken unterscheiden sich in solche, die sogenannte Wissensziele festhalten, und andere, die Änderungsziele erfassen. Wissensziele sind absolute Fakten, die evaluiert werden, wie die Güte der Produktqualität, oder der Stand des Testsaufwandes. Änderungsziele sind Messungen von Steigerungen oder Reduzierungen im Laufe der Zeit und von einer Iteration zur nächsten.

Business-Modeling

Die Modellierung der Geschäftsprozesse dient dem Verständnis der Struktur und der Dynamik einer Organisation. Das Modell stellt sicher, dass Kunden, Endanwender und Entwickler das gleiche Bild vom System erhalten. Daraus lassen sich anschließend konsistent die Systemanforderungen ableiten.

Hintergrund ist die gestiegene Funktionsvielfalt von Softwareprojekten. Im Vordergrund steht nicht eine gute Idee für einen einzelnen Algorithmus, sondern das Zusammenspiel der verschiedenen Komponenten. Software muss für einer großen Zahl von Nutzern leicht zugänglich und einsetzbar sein. Sie muss sich in der Regel in eine bestehende Software- .und Nicht-Softwarearchitektur einpassen.

Für die Modellierung werden die Diagramme der *Unified-Modelling-Language (UML)* genutzt, im besonderen *Use-Cases* und Strukturdiagramme. Der Detaillierungsgrad der Modellierung hat eine Spannweite von sehr detaillierten Abbildungen von Unternehmensmodellen bis zum Verzicht auf Business-Modeling in kleinen Projekten.

Requirements

Eine Anforderung ist eine Bedingung, der das System entsprechen muss und eine Funktionalität, die es leistet. Am Anfang stehen funktionale Anforderungen, welche definieren, was eine Software leistet. Über den Erfolg einer Software entscheiden zunehmend nichtfunktionale Anforderungen. Funktionalität selbst ist häufig in bestehenden Bibliotheken verfügbar. Ein Beispiel ist ein Packprogramm wie Winzip. Die Kompressionsfunktionalität steht in vielen Sprachen allen Programmierern zur Verfügung. Die Art der Präsentation, die Stabilität oder die Geschwindigkeit der konkreten Implementierung machen die Softwarequalität im Vergleich aus.

Nichtfunktinale Anforderungen werden nach dem *FURPS*-Modell[118] in Usability, Reliability, Performance und Supportability unterschieden. Andere Modelle wie

118 Das Akronym FURPS steht für die Begriffe Functionality, Usabilty, Reliability, Performance, Supportablity. Es wurde bei Hewlett-Packard entwickelt und von Robert Grady und Deborah Caswell erstmals 1987 veröffentlicht, siehe Grady/Caswell, Software Metrics: Establishing A Company-Wide Program.

beispielsweise die Norm DIN-66272 legen noch weitere Qualitätsmerkmale fest, wie Portability.

Da Anforderungen von Kunden – diese sind häufig nicht vom Fach – formuliert werden, besteht die Herausforderung in der Übersetzung von Prosa in Spezifikation. Zur Bildung eines gemeinsamen Verständnisses zwischen Fachabteilung und Programmierern trägt die grafische Modellierung bei.

Analysis and Design

Die wesentlichen Artefakte der Teildisziplin *Design* sind modellierte Klassen, Subsysteme und Kollaborationen. Die Aufgabe der Disziplin *Analysis and Design* ist die Übersetzung von Anforderungen in Spezifikationen, um die Brücke zwischen Anforderungsmanagement und Implementierung zu schlagen.

In der Analyse werden dabei die funktionalen Anforderungen zunächst in detaillierte *Use-Cases*, Status- und Aktivitätendiagramme übersetzt. Anschließend passt das Design die Analyseergebnisse an die Beschränkungen der Implementierungsumgebung an und berücksichtigt zusätzlich die nichtfunktionalen Anforderungen. Das Design ist eine Verfeinerung der Analyse. Das Ergebnis der Tätigkeiten ist ein Designmodell. Gelegentlich wird zur Darstellung eines abstrakten Systemüberblicks ein separates Analysemodell vorgehalten

Der Arbeitsablauf *Analysis and Design* ist also ein zweistufiger Prozess. Die Durchgängigkeit von Anforderung, zum Analysemodell, zum Designmodell bis in den Code ist unterschiedlich. Kann aus den Klassendiagrammen direkt der Code generiert werden, wird dies als *Round-Trip-Engineering* bezeichnet.

Implementation

In der Disziplin *Implementation* findet die eigentliche Programmierung statt. Nach der Philosopih des RUP ist Softwareentwicklung mehr als reine Programmierung. Dennoch bleibt die Programmierung natürlich die Kerndisziplin. Zur Programmierung gehört die Organisation des Quelltextes in Schichten und Subsysteme. Das Konzept der Objektorientierten Programmierung (OOP) ist implizite Voraussetzung. Zur Disziplin *Implementation* gehört die Erstellung von automatisierten Komponententests und die Organisation der Integration der Softwarekomponenten.

Zu jeder Softwarekomponente gehört mindestens ein Komponententest. Die Verantwortung für die Konzeption und Umsetzung dieser Tests liegt beim Entwickler. Es erfolgt eine Abgrenzung zu System- und Komponententests, diese werden in der Disziplin Test konzipiert und durchgeführt.

Ein wesentliches Konzept von iterativen Vorgehensmodellen sind die Vorgänge des *Build* und der *Integration*. Die in die Versionsverwaltung gegebenen Quelltexte sind stets kompilierbar. Dies wird nach dem Konzept der kontinuierlichen Integration in regelmäßigen Abständen zu einer Erzeugung des Softwareproduktes genutzt. Es finden sogenannte *Daily-Builds* Anwendung, beziehungsweise aufgrund der besseren Praktikabilität *Nighly-Builds*.

Unter einer Integration wird der Zusammenbau des kompletten Systems zu einem ausführbaren Release verstanden. Nach dem iterativen Ansatz des RUP steht am Ende jeder Iteration ein ausführbares Release. So können die geplanten Features getestet und darauf aufbauend die nächste Iteration geplant werden.

Prototypen

Ein wichtiges Konzept der Disziplin *Implementation* ist das *Prototyping*. Darunter wird die Programmierung eines Ansatzes verstanden, der die Realisierbarkeit oder das Verhalten von Kernelementen ausprobieren möchte. Ziel kann die Erforschung der Stabilität oder der Performance einer Schlüsseltechnologie sein. Oder der Prototyp kann als Anschauungsobjekt zur Verdichtung von Anforderungen in der Absprache mit dem Kunden dienen. Wesentlicher Grund für die Investition in Prototypen ist Risikominimierung. Es ist besser, wenn ein Projekt zusätzlich Zeit in die Realisierung eines Prototypen steckt, als spät im Projekt festzustellen, dass eine Schlüsseltechnologie nicht beherrschbar ist oder eine abstrakt formulierte Anforderung den Kundenvorstellungen nicht entsprach. Es können zwei Typen unterschieden werden:

- **Exploratory-Prototype:** Diese werden nach dem Gewinn der Erkenntnisse und deren Dokumentation verworfen. Exploratory Prototypes werden mit minimalem Aufwand entwickelt und arbeiten ohne durchdachte Architektur. Häufig kommen fremde Sprachen oder Technologien zu Einsatz, wie eine nicht benutzte Programmiersprache, die aber den persönlichen Fähigkeiten oder dem Geschmack eines Entwicklers näher kommt.

- **Evolutionary-Prototype**: Diese werden von vornherein so konzipiert, dass sie den Projektstandards entsprechen und können daher im Laufe der Entwicklung zum finalen System weiterentwickelt werden.

Wichtige Voraussetzung an *Prototyping* ist, dass die Art des Prototyps **vor** seiner Umsetzung feststehen muss. Bei der Entwicklung eines *Exploratory-Prototype* zum finalen System können sich im Kern später nur sehr schwer zu identifizierende Fehlerquellen einschleichen. Ein weiteres Unterscheidungsmerkmal ist das Ziel, welches mit dem Prototypen erreicht werden kann:

- **Behavioral-Prototype:** Diese versuchen darzustellen, wie sich das System aus Sicht des Nutzers verhält und nehmen keinen Bezug auf die Architektur des Systems. Aus der oben getroffenen Sicht auf das Ergebnis des Prototyps sind diese Prototypen häufig *Exploratory Prototypes*, da sie nur einen kurzen Eindruck vom Verhalten liefern. Eine Webseite kann jedoch auch zunächst als Prototyp nur ein Verhalten dokumentieren und im und im Folgenden auf dieser Basis zur fertigen Anwendung entwickelt werden.

- **Structural Prototypes:** Diese neigen dazu, die Infrastruktur des fertigen Systems zu nutzen. Diese Art von Vorgehen wird auch „Durchstich" genannt und hat zum Ziel in einer Art Smoke Test überhaupt erst einmal

ein Ergebnis oder Teilergebnis zu erzielen, um eine grundsätzliche Realisierbarkeit beweisen zu können. Von ihrer Art her sind diese Prototypen häufig Evolutionary Prototypes. Sie müssen es jedoch nicht sein.

Der *Rational-Unified-Process* empfiehlt die Nutzung eines evolutionären, strukturellen Prototyps, welcher im Laufe der Phasen und Iterationen zum fertigen System entwickelt wird. Parallel dazu sollten immer wieder Wegwerfprototypen zur Dokumentation des Systemverhaltens eingesetzt werden.

Test

Der Zweck des Testens liegt in der Bewertung der Softwarequalität. Der Maßstab guter Qualität wird durch die Anforderungen festgelegt. Die Qualität überprüfen bedeutet sicherzustellen, dass alle Anforderungen an die Software erfüllt sind. Weiterhin ist es fester Bestandteil des RUP, dass die Software in einem von allen Beteiligten anerkannten Prozess entwickelt wurde. Daher beinhaltet die Qualität auch das Finden von Kriterien und Verfahren[119], um sicherzustellen, dass das Produkt auch die Anforderungen erfüllt. *Test* umfasst also Produkt- und Prozessqualität.

Die Disziplin *Test* beinhaltet außerdem den gesamten Bereich des Testmanagements, welcher eigene Bücher füllt. Die Theorie unterscheidet verschiedene Testmöglichkeiten in feingranularer Form, beispielsweise in Zeitpunkte (Smoke Tests, Einzeltests, Integrationstests, Systemtests oder Akzeptanztest) oder in Testarten (Benchmarks, Funktionstests, Streßtests und ähnliche). Die Beschreibung dieser Vielfältigkeit würde diesen Rahmen sprengen[120]. Wichtig für die Disziplin *Test* im *RUP* ist neben den genannten Kriterien die Tatsache, dass die Qualitätssicherung während des gesamten Entwicklungsprozesses arbeitet, und damit insbesondere dem Ansatz Rechnung trägt, dass jede Iteration des *RUP* ein ausführbares Release hervorbringt.

Configuration- and Change-Management

Die Disziplin *Configuration and Change Management* hat in beiden Ausprägungen – Konfiguration und Veränderung – die Aufgabe die Integrität der produzierten Artefakte sicherzustellen, denn sie stellen einen erheblichen Investitionswert dar.

Das *Configuration-Management* unterzieht die Artefakte einer Versionskontrolle. Es muss in der Lage sein die Artefakte, ihre Versionen und die Änderungshistorie darzustellen und zu rekonstruieren. Weiterhin werden die Abhängigkeiten zwischen Artefakten behandelt, die Konsistenz zwischen generierten Dokumenten wie Binaries oder PDF-Dokumenten und ihren Quellen wird sichergestellt. Die Erkenntnisse über Abhängigkeiten steuern das *Build-Management*.

---

119 Siehe Kruchten , The Rational Unified Process, S. 193ff.

120 Siehe beispielsweise in inhaltlicher Übereinkunft mit dem Glossar des International Software Testing Qualifications Board: Spillner/Linz, Basiswissen Softwaretest oder das TMap-Modell in Koomen/van der Aalst/Broekman/Vroon, TMap Next – Ein praktischer Leitfaden für ergebnisorientiertes Softwaretesten.

Das *Change-Management* hängt mittelbar mit dem *Configuration-Management* zusammen. In iterativen Projekten sind Änderungen an den Anforderungen, der Architektur und der Technik ausdrücklich erwünscht.

Aufbauend auf der Integrität der Artefakte können Status und Messungen entnommen werden, die dem *Project-Management* als Grundlage für Metriken dienen. Eine Aufgabe des *Configuration-Management* ist es häufig, den Einsatz von Werkzeugen zur statischen Code-Analyse oder ähnlichen Qualitätssicherungswerkzeugen zu konfigurieren.

Environment

Mit *Environment* ist die Softwareentwicklungsumgebung gemeint. Die Disziplin setzt diese Umgebung jedoch nicht nur auf, sondern definiert sie und leistet technischen Service. Einige Aufgaben von *Environment* hängen eng mit dem *Configuration-Management* zusammen. Während es dort jedoch eher um die Möglichkeiten und Vorteile der Werkzeuge geht, steht hier die eigentliche Administration im Vordergrund. Kleine Projekte können die Aufgaben in Personalunion betreuen.

Deployment

Im Deployment wird die Auslieferung der Software an den Kunden bearbeitet. Dies kann eine Fülle verschiedener Aufgaben zum Inhalt haben, je nach Typ des Produktes, wie bei der Beschreibung der *Transition-Phase* beschrieben. Wichtig ist die Erkenntnis der Orthogonalität der Disziplin *Deployment* und der Phase *Transition*. Denn die Disziplin ist iterativ am ganzen Entstehungsprozess der Software beteiligt, insbesondere ist es erstrebenswert in der kontinierlichen Integration dieselben Umgebungen nutzen zu können, wie bei der Integration eines Realease. *Deployment* behandelt alle Artefakte, die ausgeliefert werden, also Software, Datenbanken, Installationsskripte, Dokumentation. Es kann ebenfalls weitere Unterstützung der Endanwender mit umfassen, also Maßnahmen wie Schulungen, Telefon- und Internetsupport, die ebenfalls in der Disziplin Deployment konzipiert und ausgeführt werden.

4. Werkzeuge

Das Softwareprojektmanagement kommt gerade bei Anwendung eines Vorgehensmodells nicht ohne den Einsatz spezialisierter Softwarewerkzeuge aus. Philippe Kruchten [P03] benennt diese Erkenntnis als eine der wichtigsten Eigenschaften des Rational Unified Process:

„To be effective, a process must be supported by adequate tools. The Rational Unified Process is supported by a vast palette of tools that automate steps in many activities. These tools are used to create and maintain the various artifacts – models in particular – of the software engineering process, namely, visual modeling, programming, ans testing.[121]“

Für den *RUP* haben Werkzeuge immanente Bedeutung, denn er wird von der Firma IBM/Rational gemeinsam mit einer Palette von Werkzeugen vertrieben. Die Verknüpfung von Prozess und Werkzeug ist ein Geschäftsmodell[122]. Letztlich ist die Formulierung des RUP mit seinem Bezug zur UML erst mit der Entwicklung leistungsfähiger Modellierungswerkzeuge möglich geworden. Auch für die erfolgreiche Durchführung von anderen Vorgehensmodellen ist eine spezialisierte Werkzeugpalette notwendig; es ist daher in den vergangenen Jahren ein Markt dafür entstanden. Die Hersteller versuchen mit ihren Konzepten möglichst viele verschiedene Vorgehensmodelle zu bedienen.

Aufgrund der Verwebung mit dem Prozess selbst werden im Folgenden die aus der Sicht des *RUP* notwendigen Werkzeuge vorgestellt. Es werden die zugehörigen Produkte der Firma IBM/Rational[123] genannt. Es besteht jedoch keine Notwendigkeit nur die Produkte dieser Firma zu verwenden. Aufgrund der hohen Lizenzkosten haben sich etliche Alternativen auf dem Markt etabliert, die entweder günstiger oder kostenlos sind, einen etwas veränderten Modellansatz vertreten oder sich schlichtweg als überlegene Produkte (technisch, im Hinblick auf Schnelligkeit, Stabilität, Benutzbarkeit) verkaufen wollen.

Nach Kruchten sind für die Durchführung von Softwareprojekten folgende Werkzeuge notwendig[124].

4.1 Werkzeuge zur Modellierung

Damit ist ein UML-Modellierungswerkzeug gemeint. IBM bietet zu diesem Zweck *Rational Rose* an. Dieses ist in der Lage *Round-Trip-Engineering* zu betreiben, also die automatische Generierung von Code aus einer objektorientierten Designspezifikation und umgekehrt. Modellierungswerkzeuge werden vornnehmlich von der Disziplin *Analysis & Design* eingesetzt, aber auch in der Geschäftsmodellanalyse und dem Anforderungsmanagement.

121 Siehe Kruchten, The Rational Unified Process, S. 30ff.
122 Siehe dazu Bunse/von Knethen, Vorgehensmodelle kompakt, 2. Auflage, S. 55ff.
123 Nach der Übernahme von Rational durch IBM im Jahr 2003 ist Rational das Markenzeichen unter dem der RUP und die Werkzeugpalette vertrieben und weiterentwickelt werden.
124 Siehe Kruchten, Der Rational Unified Process – Eine Einführung, S 197f.

Nachdem modellbasierte Softwareentwicklung auch Teil anderer moderner Vorgehensmodelle geworden ist, entstand ein Markt für Modellierungswerkzeuge. Solche Werkzeuge heißen *UML*-Werkzeuge oder allgemeiner auch *CASE*-Tools[125], wenn sie neben der *UML* auch weitere grafische Notationen beherrschen. Bekannte Vertreter sind *Enterprise Architect* von Sparx Systems und *Together* von Borland. Auf dem freien Markt existieren *Umbrello*, *StarUML* mit eingeschränktem Funktionsumfang und natürlich *Fujaba,* entwickelt an der Universität Paderborn.

Andere Produkte werden als Plug-In für die freie Entwicklungsumgebung *Eclipse* entwickelt. Dazu gehört das Produkt *OmondoUML* der Firma Omondo, von welchem es eine kostenlose Variante gibt. Hoffnungsverheißend ist das *Eclipse Modeling Framework (EMF)*, welches *Eclipse* selbst Modellierungsfähigkeiten geben soll.

Insgesamt ergibt sich ein ambivalentes Bild. Auf der einen Seite stehen sehr leistungsfähige Modellierungssuites wie *Rose* und *Togother*, die aber gleichzeitig auch hohe Lizenzgebühren kosten und auf der anderen Seite noch nicht komplett ausgereifte Ansätze des freien Marktes. Einen Mittelweg geht der nicht freie, aber vergleichsweise günstige *Enterprise Architect*.

4.2 Anforderungsmanagement

Es ist die besondere Eigenschaft von Anforderungen als Schnittstelle zwischen Auftraggeber und Auftragnehmer, welche eine besondere Aufbereitung der Daten notwendig werden lassen. Eine rein textuelle Beschreibung (ein Word-Dokument) ist oft missverständlich und kann gleichzeitig von der IT-Seite nicht effektiv verarbeitet werden. Eine Lösung besteht in einer Mischung aus (abstrakten) Modellen und Text. Rational sieht für die Verwaltung von Anforderungen das Produkt *RequisitePro* vor. Es kommt aber zur Modellierung auch *Rational-Rose* zum Einsatz. Das Produkt *Rational-SoDA* kann dann aus den Modellen und dem Text eine geschlossene Darstellung in verschiedenen Formaten generieren.

Ob dieses Vorgehen der Weisheit letzter Schluss ist, kann noch nicht entschieden werden. Gerade im Bereich des Anforderungsmanagements existieren viele weitere sehr unterschiedliche Ansätze. Einige fachfremde Werkzeuge versuchen das Anforderungsmanagement zusätzlich zu erfassen, so bietet *Enterprise Architect* auch ein Meta-Modell zur Erfassung von Anforderungen. Einige Test-Tools bieten zur einfachen Verknüpfung mit Testfällen ebenfalls eine solche Funktion, das Produkt *Quality-Center* von Hewlett-Packard ist nur ein Beispiel dafür. Die Firma Gebit Solutions prägte mit Ihrem Produkt *Trend/Analyst* die These vom *Model-Driven-Requirements-Engineering (MDRE)* und stellt eine Lösung bereit, welche nahtlos eine UML-Teilmenge und Text im Stil eines Wikis integrieren kann. Und letztlich hat IBM selbst mit dem Zukauf von *Telelogic* samt des Flagschiffs *Doors*, einem auf Anforderungsmanagement spezialisierten

125 Der Ausdruck CASE steht für Computer-Aided-Software-Engineering. Computerunterstützte Softwareentwicklung muss sich nicht nur auf grafische Modellierung beschränken, auch integrierte Entwicklungsumgebungen mit Autovervollständigung und ähnlichen Konzeoten gehören dazu. Sie dazu Wikipedia unter dem Stichwort Computer-Aided-Software-Engineering.

Werkzeug, Handlungsbedarf signalisiert.

Ursache für diese Bewegungen ist sicherlich die Erkenntnis des Scheiterns vieler Softwareprojekte am mangelnden Anforderungsmanagement. Besondere Schwierigkeiten treten in der Darstellung auf, weil diese von IT-Spezialisten genauso wie von fachfremden Kunden verstanden werden muss. Welcher Ansatz sich durchsetzen wird ist nach heutigem Stand offen. Klar scheint nur zu sein, dass die Darstellung von Anforderungen in einem einfachen Text- oder Tabellenformat ausgedient hat.

4.3 Codeentwicklung

Keine Software wird heutzutage nur in einem Editor geschrieben. Mit *Eclipse* steht eine freie und äußerst leistungsfähige integrierte Entwicklungsumgebung bereit, welche mittlerweile weit mehr als nur die Java-Plattform unterstützt. Dies hat wohl auch Microsoft auf den Plan gerufen. Für die Sprachen C#, C++, Visual Basic und für die Webentwicklung werden sogenannte Express-Versionen des *Visual Studio* frei auch für den kommerziellen Einsatz bereitgestellt. IBM bietet neben der finanziellen und ideellen Unterstützung von *Eclipse* den *Rational-Application-Developer* an, ein *Eclipse*-Derivat, angereichert mit Plug-Ins zur Entwicklung von großskalierenden Webanwendungen.

Weiterhin gehören zur Codeentwicklung nach Rational auch die Produkte der *Purify*-Serie, welche Werkzeuge zum Profiling (*Rational Quantify*), zur Überprüfung der Codeabdeckung (*Rational PureCoverage*) und der Speicherverwaltung (*Rational Purify*) umfasst. Diese kommen in den Disziplinen *Implementation* und *Test* zum Einsatz.

Für diese Zwecke gibt es zusätzlich einen großen Markt von freien und kommerziellen Produkten, so dass die *Rational*-Palette in diesem einen kleinen Marktanteil hat.

4.4 Konfiguration- und Änderungsmanagement

Für das Konfigurationsmanagement wird vorallem eine Versionsverwaltung benötigt. Rational bietet dazu *ClearCase* an. Traditionell sind aber auch freie Produkte für diesen Einsatzzweck bekannt, bereits seit 1989 existiert das *Concurrent Versions System (CVS)* mit weiter Verbreitung gerade im Open-Source-Markt. *Subversion* wurde mit dem Ziel entwickelt, einige Schwächen von *CVS* zu beheben (und insbesondere Kopien und Verschiebungen von Dateien ohne Verlust der Versionshistorie zu ermöglichen) und es abzulösen.

Für das Änderungsmanagement steht mit *ClearQuest* ein *Bug-Tracking-Tool* bereit. Für diesen Einsatzzweck gibt es neben dem verbreiteten freien Werkzeug *BugZilla* einige weitere webbasierte Angebote. Das produkt *Jira* von Atlassian hat aufgrund seines Workflow-Charakters Verbreitung gefunden. Aber auch die Werkzeuge aus dem Bereich des Testmanagements sind häufig in der Lage eine Fehlerverwaltung zu führen, so zum Beispiel das HP's *Quality Center*.

4.5 Test

Die Bedeutung des Testmanagements wird zunehmend größer, dies zeigt die Gründung der ISTQB [O14] und die Zunahme von Fachartikeln zum Thema in diversen Zeitschriften. Konsequenterweise tut sich auch ein eigener Markt für spezialisierte Testwerkzeuge auf. Im Bereich der funktionalen Tests wird ein Werkzeug zur Verwaltung der Testfälle und die Protokollierung der Testausführung erforderlich. Im Bereich der nicht-funktionalen Tests kommen Werkzeuge zur Performanzmessung und für Last- und Streßtests zum Einsatz. Es werden für die Testautomatisierung Werkzeuge mit unterschiedlichen Ansätzen eingesetzt, besonders im Webumfeld häufig sogenannte Capture-Replay-Werkzeuge. Auch in nicht-agilen Projekten sind Unit-Tests mit der Unterstützung eines *jUnit*-Derivats immer sinnvoll.

Die Werkzeuglandschaft im Bereich Test ist also vielfältig. Entsprechend bieten die Werkzeughersteller komplette **Tool-Suiten** an. Marktführer sind die Werkzeuge von *Hewlett Packard*, das *Quality Center*, der GUI-Automatisierer *WinRunner* und *LoadRunner* für Last- und Performance-Tests. *Rational* bietet eine ähnliche Palette, genauso Borland mit den *Silk*-Tools. Daneben existieren freie Werkzeuge für einzelne Aspekte des Tests, beispielsweise *FitNesse* (GUI-Automatisierung) vom Ward Cunningham oder *Selenium* zum Test von Webanwendungen.

4.6 Planung und Verfolgung

In der Übersicht von notwendigen Softwareprodukten erwähnt Philippe Kruchten explizit Werkzeuge zur „Planung und Verfolgung“, allerdings ohne ein konkretes Beispiel zu nennen. Im Kern scheint es um die Anforderung zu gehen, den Prozessvorgang *an sich* überprüfbar zu machen. Die *HTML*-Oberfläche des *RUP* bietet dazu *Word*-Vorlagen an. In vielen Projekten kommt *Project* von Microsoft zum Einsatz.

Insgesamt scheinen damit einerseits die klassischen Instrumente des Projektmanagements in Form von Gantt-Diagrammen zum Einsatz zu kommen. Andererseits bildet sich ganz aktuell eine neue Sorte von Projektmanagementwerkzeugen heraus; Werkzeuge für das sogenannte *Application-Lifecycle-Management (ALM)*. Unter *Application-Lifecycle* wird dabei der Lebenszyklus einer Anwendung während ihrer Entwicklung verstanden. *ALM*-Tools sorgen also für das Management der gesamten Werkzeug- und Aktivitätenketten während der Entwicklung. Sie gehen damit insbesondere auf die Erfordernisse der neuen, ausgereiften Vorgehensmodelle ein, denen die klassischen Instrumente wie *Project* allein nicht mehr genügen. Die konkreten Ansätze sind dabei durchaus unterschiedlich und orientieren sich an den Vorgehensmodellen.

Die Firma Serena [O15] beispielsweise vermarktet ihre sogenannten *Business-Mashups*, eine Möglichkeit Arbeitsabläufe in einem Werkzeug abzubilden und Aufgaben, Tätigkeit und Weiteres agil bestimmten Projektrollen zuzuweisen und ausfüllen. Einen integrierten Ansatz bietet die Firma Polarion [O16] mit webbasierten Lösungen. Besonders interessant ist die Aufstellung der Firma Borland. *Borland* ist in der IT-Welt bekannt für die Herstellung von IDE- und

Compilertechnologien. Es wissen aber wenige, dass *Borland* diese Sparte im Jahr 2006 verkauft hat und aktuell unter der Marke *Borland – The Open ALM Company* ausschließlich *ALM*-Produkte verkauft. Entsprechend großes Potential müssen die Entscheidungsträger in diesem Markt erkennen.

4.7 Aufbereitung von Dokumentation

Kruchten betont die Notwendigkeit von Werkzeugen zur Aufbereitung von Dokumentationen. Konkret behandelt er allerdings nur das Produkt *Rational-SoDA*. Ein solches Produkt wird bei integrierten Konzepten obsolet. Die Erwähnung dieses Punktes verleiht aber einem Wunsch Ausdruck, welcher unabhängig vom Vorgehensmodell jedem Projekt innewohnen dürfte. Es ist trotz spezialisierter Werkzeuge für die Arbeitsdokumente notwendig, geschlossene Dokumentationen zu erzeugen. Diese haben in einigen Anwendungsbereichen gar Urkundencharakter, beispielsweise in den kontrollierten Märkten der medizinischen Geräteherstellung oder im Bankensektor. Diesem Anliegen tragen viele Werkzeuge mit Exportschnittstellen nach HTML, PDF oder DOC Rechnung. Letztlich bedeutet dies aber auch, dass die klassische Textverarbeitung auch angesichts der Masse spezialisierter Werkzeuge natürlich nicht aus einem Projekt wegzudenken ist. Neben der *Office-Suite* von Microsoft hat sich dafür auch *OpenOffice* zu einer festen Größe etabliert.

Teil V Praxisbericht

Das Projekt wird nach dem Modell des *Rational Unified Process* durchgeführt. Die Arbeitsschritte und die Dokumentation folgen der Aufteilung in die Phasen und die Disziplinen des *RUP*. Wegen seiner Eigenschaft als akademisches Projekt soll die Denkweise in Rollen verstärkter Anwendung finden als dies für den praktischen Einsatz notwendig wäre. Das Projektmitglied soll also in der Rolle als Anforderungsmanager andere Ziele verfolgen als in der Rolle des Entwicklers. Gleichzeitig muss die Dokumentationslast auf notwendige Dokumente reduziert werden, um die Arbeitslast nicht übermäßig zu erhöhen. Es wird aber davon ausgegangen, dass gewisse Dokumente, insbesondere Modellierungsdokumente das bekannte *„Trial&Error“*-Prinzip weitgehend ersetzen können und die Modellierungsaufwände in Vergleich zu Redundanzen während der Entwicklung geringer sind.

1. Inception-Phase

Die Inception-Phase dient der Konzeptfindung. Für das konkrete Projekt liegt eine klare Aufgabenbeschreibung vor, es sind keine Vertragsverhandlungen notwendig. Daher kann die *Inception-Phase* verhältnismäßig kurz gestaltet sein, geplant ist lediglich eine Iteration. Im Folgenden werden einige Aktivitäten der Disziplinen *Requirements*, *Business-Modeling* und *Project-Management* beschrieben.

Requirements

Der Auftraggeber hat zur Ausschreibung des Projektes ein Lastenheft erstellt. Demnach ist ein Abstimmungssystem in Form einer Webapplikation zu erstellen, welches das bisherige Umlaufsystem für Distanzabstimmungen – durchgeführt über gedruckte oder elektronische Post – ablösen soll. Neben einer Reihe von funktionalen Anforderungen in Bezug auf Eingabe- und Ausgabeformate, beschreibt das Lastenheft Akteure; im Detail einen Administrator der Abstimmung, den „Vorsitzenden“, und Abstimmungsteilnehmer, die stimmberechtigten „Mitglieder eines Gremiums“. Die Identifikation der Mitglieder erfolgt über die dem Administrator bekannten E-Mailadressen. Dazu korrepondieren die Schnittstellen zur Benutzung des Systems durch die Akteure. Dies ist zum einen eine administrative Schnittstelle – Schnittstelle A – und eine Schnittstelle zur Abstimmung – Schnittstelle V (Vote). Daraus ergeben sich direkte Arbeitsabläufe, *Use-Cases* in der Nomenklatur des *RUP*. Der genaue Wortlaut des Lastenheftes befindet sich im Anhang.

Aus den Anforderungen des Lastenheftes wird von der Disziplin *Requirements* ein Pflichtenheft erarbeitet. Die dort erfassten Anforderungen werden in tabellarischer Form geführt. Jede Anforderung erhält eine eindeutige Identifikationsnummer, eine Beschreibung und eine Priorität. Zusätzlich wird die Komponente genannt, auf welche sich die Anforderungen bezieht. Die Ergebnisliste ist das Pflichtenheft für das Projekt. Sie befindet sich ebenfalls im Anhang.

Die Anforderungen werden aus dieser Liste in Form von Änderungsanträgen weiterentwickelt. Die Disziplin *Business-Modeling* nutzt die Anforderungen für eine graphische Modellierung der Geschäftsprozesse.

Business-Modeling

Zur Darstellung der Geschäftsprozesse werden *Use-Cases* in UML modelliert. Die Grafiken werden zur besseren Darstellung um weitere Strukturierungselemente wie Rahmen und Bezeichnungen erweitert.

Der Basisworkflow beschreibt eine direkte Umsetzung der Anforderung in eine Abfolge von Arbeitsschritten, die mit dem System vollzogen werden können.Abbildung 4 zeigt die Behandlung einer Abstimmung und mit den beiden Komponenten `Admin` und `Vote`. Das *Use-Case*-Diagamm kennt zwei Akteure, den Administrator und den Stimmberechtigten. Es ergeben sich im Kern folgende

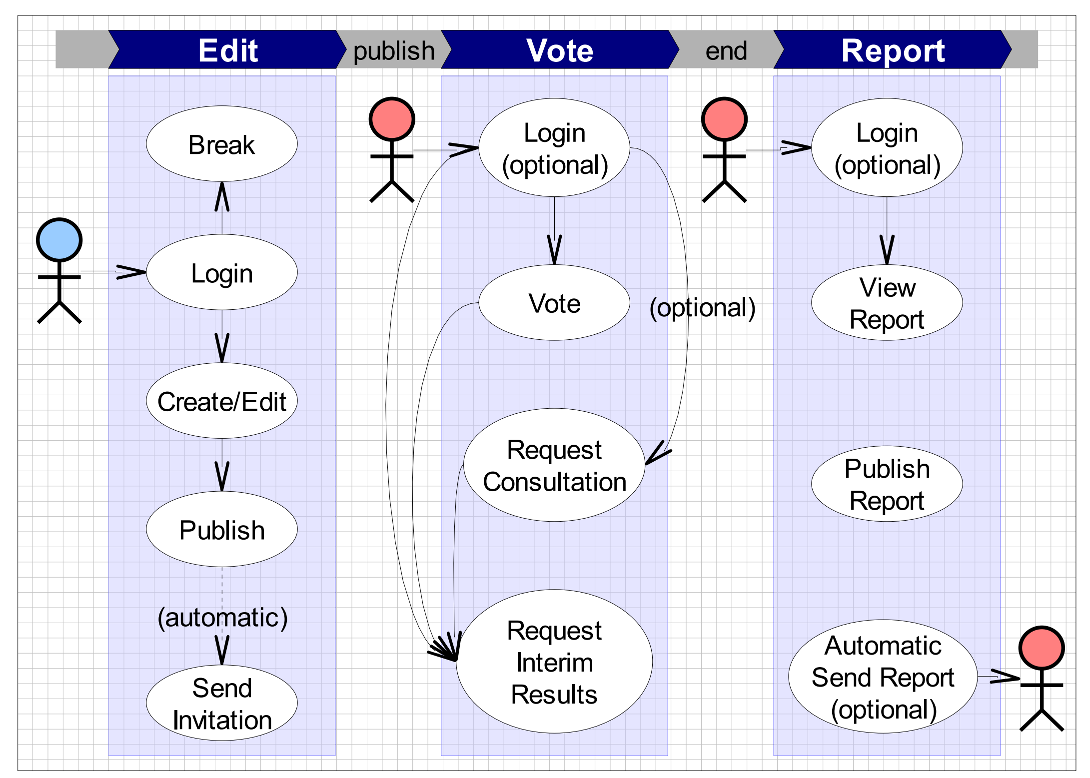


Abbildung 4: Darstellung der Schritte zur Systemnutzung

vom System abgedeckte Anwendungsfälle[126]:

1. Die Erstellung einer neuen Abstimmung (CREATE). Dazu ist ein Besuch der Startseite, das Anmelden mit Benutzernamen und Passwort und schließlich das Durchlaufen der einzelnen Schritte zur initialen Erstellung einer Abstimmung erforderlich.
2. Bearbeiten einer Abstimmung (EDIT). Erfolg analog zur Erstellung. Voraussetzung ist, dass der Nutzer Administrator der zu bearbeitenden Abstimmung ist.
3. Betrachten eines Ergebnisses oder Zwischenergebnisses (VIEW). In der ersten Version der Software kann jede Abstimmung von jedem Benutzer betrachtet werden. Nur der Vorgang der Stimmabgabe selbst wird durch das Rechtekonzept reglementiert.
4. Abstimmen (VOTE). Nachdem ein Benutzer die Information über eine ihm mögliche Abstimmung erhalten hat, geht er auf die Startseite, meldet sich am System an (sofern erforderlich), sucht die Abstimmung über die Suchmaske und stimmt ab.

Project-Management

In der Disziplin *Project-Management* müssen in der ersten Projektphase viele

126 Die Darstellung stammt aus einer frühen Phase des Projektes und zeigt die Weiterentwicklung der Anforderung in ein Geschäftsmodell. Das Modell zeigt aber nicht den letzten Stand, es wurde im Laufe des Projektes ebenfalls weiterentwickelt.

strategische Entscheidungen über die Planung und Organisation des Projektes getroffen werden. Diese Dinge werden hier nicht näher beschrieben. Weiterhin wird in dieser Phase das Visionsdokument entworfen[127].

Corporate Identitiy

Die Webanwendung wird von keinem aggressiven CI geprägt, es soll sich produktiv in unterschiedliche Intranetumgebungen einbinden lassen. Der Name des Projekte ist ***rogatio*****.net**. Rogatio ist eine lateinische Vokabel für „Frage“ und weiterhin ein Begriff aus dem römischen Recht mit dem ein Antrag an die Legislative gemeint ist. Der zweite Teil „Dotnet“ deutet auf die Eigenschaft als Webapplikation hin. Letztlich wird mit dem Namen als Fernziel eine Veröffentlichung der Anwendung unter der Webadresse http://www.rogatio.net avisiert.

Andere Disziplinen

In den anderen Disziplinen fallen in der *Inception-Phase* wenige Aufgaben an. In einem realen Projekt hängt deren Auslastung vom verfügbaren Personal ab. Wenn schon Testmanager, Konfigurationsmanagemer, Entwickler oder Architekten im Projekt arbeiten, so würden auch sie mit ersten Konzepten beginnen. Diese Arbeit legt einen Schwerpunkt auf konkreten Entwürfen, die erst in der *Elaboration-Phase* final sind. Diese beruhen dann auf Konzepten, deren Darstellung hier ausgespart wird.

127 Die Inhalte des Visionsdokumentes werden nicht wiedergegeben, weil sie im Wesentlichen der Darstellung des abgedruckten Handbuches entsprechen.

2. Elaboration-Phase

Aufbauend auf den Konzepten aus der *Inception-Phase* werden in der *Elaboration-Phase* Entwürfe erstellt. Dies betrifft insbesondere die Disziplinen *Analysis & Design* und *Implementation*. Im Projektmanagement müssen weitere richtungsweisende Entscheidungen getroffen werden. Darauf aufbauend beginnt die Disziplin *Environment* mit dem Aufbau der Entwicklungsumgebung. Spätestens zu diesem Zeitpunkt sollten die Disziplinen *Test* und *Configuration-Management* ihre Tätigkeiten aufnehmen.

Für diese Arbeit sind die Ergebnisse der *Elaboration-Phase* von zentraler Bedeutung, weil sie die Grundideen des Projektes darstellen. Im Folgenden werden daher einige prinzipielle Tätigkeiten und die wichtigsten Artefakte dieser Phase vorgestellt.

2.1 Iteration E1

Die Elaboration-Phase gliedert sich in zwei Iterationen E1 und E2. In die erste Iteration werden wichtige Konzepte konkret dargestellt. Die zweite Iteration verarbeitet die Erkenntnisse und passt einige Entscheidung an. Dies zeigt den Charakter eines iterativen Vorgehensmodells.

Analysis & Design

Screens

Die Anforderungen an den Workflow lassen sich in neun Screens abbilden. Ein Screen ist dabei die geschlossene Darstellung eines gewissen Funktionsrahmens.

1. **Startseite**. Notwendig ist zunächst eine Startseite. Hier stehen neben einführenden Worten Listen mit den neuesten Abstimmungen oder Abstimmungen, die bald enden.

2. **Anmelden**. In der ersten Version der Software wird es bei jedem Besuch der Anwendung notwendig sein, sich anzumelden[128].

3. **Abmelden**. Das System bietet die Option einen Benutzer wieder abzumelden[129].

4. **Registrieren**. Zur Einrichtung eines neuen Benutzers gibt es einen eigenen Bildschirm zur Registrierung.

5. **Erstellen**. Nach der Anmeldung ist das Erstellen einer neuen Abstimmung möglich. Die vielfältigen Daten werden in mehreren Schritten erfasst, alle

128 Diese Modellierung wird im Verlauf des Projektes verändert. Mit der Einführung der HTTP-Authentifizierung erfolgt eine Anmeldung am System erst bei Notwendigkeit.

129 Diese Anforderung wird ebenfalls geändert werden. Das Konzept der HTTP-Authorisierung sieht keine Abmeldung vor.

Dialoge gehören aber zum gleichen Screen (mehrfache Aufrufe der gleichen Seite mit anderen Parametern).

6. **Bearbeiten**. Eine Anwendung kann vor dem Start von Abstimmungen vom Administrator bearbeitet werden.
7. **Suchen**. In einer Suchmaske werden Filterkriterien für die Auflistung von Abstimmungen definiert.
8. **Betrachten**. In einem Screen werden die Details einer Abstimmung, ihr Status und das Zwischen- beziehungsweise Endergebnis dargestellt.
9. **Abstimmen**. In einem Eingabedialog wird berechtigten Nutzern die Möglichkeit zur Abgabe einer Stimme gegeben.

Pages

Die einzelnen Screens sind auf verschiedene physikalisch im Dateisystem vorhandene Seiten der Webanwendung verteilt. Die Screens Anmeldung und Abmelden sind auf einer Seite zusammengefaßt, da nur abhängig vom jeweiligen Zustand (angemeldet oder abgemeldet) die beiden Optionen angezeigt werden müssen. Die Screens zum Erstellen und Bearbeiten werden ebenfalls von einer Seite verarbeitet. Das Suchen ist verteilt auf zwei Seiten Eingabe und Ausgabe. Alle anderen Screens entsprechen genau einer Seite.

Menüführung

Nicht aus jeder Seite ist der Übergang zu allen anderen Seiten möglich. Eine Abstimmung ist beispielsweise nur aus der Anzeige von Details möglich. Bestimmte Übergänge sind nur im Falle eines angemeldeten Benutzers möglich.

Schichtenmodell

Aus dem Lastenhaft der geplanten Anwendung lässt sich bereits ein Systemschwerpunkt in der Datenhaltung ableiten. Abstimmungen werden angelegt, verändert und anschließend von Stimmberechtigten mit weiteren Daten befüllt. Es sind zunächst zwei Komponenten sichtbar, nämlich eine Schnittstelle zur Administration eines Abstimmungssystems (Komponente Admin) und eine Schnittstelle zur Durchführung der Abstimmung (Komponente Vote). Alle Daten werden persistiert und zustandslose Webseiten liefern eine Sicht auf diese Daten. Technisch gesprochen erfüllen die beiden Komponenten also drei Aufgaben:

1. Das Auslesen und Speichern der Daten.
2. Die Generierung von Sichten auf diese Daten und eine Ablaufsteuerung in Bezug zu den übergebenen Parametern.
3. Die Darstellung und Ausgabe in einem browser-lesbaren Format.

Daher basiert die Systemarchitektur auf dem Pattern Model-View-Controller[130].

130 Siehe zum Pattern Model-View-Controller beispielsweise Freeman/Freeman, Head First Design Patterns, S. 526ff.

Abbildung 5zeigt eine grafische Übersicht. Der Bereich *View* wird von drei Komponenten gebildet. Neben den im Anforderungsdokument genannnten Komponenten *Admin* und *Vote* wird an dieser Stelle noch eine eine Komponente *Setup* hinzu. Diese übernimmt Installationprozesse, die nach der aktuellen Planung ebenfalls webbasiert durchgeführt werden können.

Festgesetzt wird, dass die Darstellungskomponenten tatsächlich nur die Aufbereitung in Hypertext vornehmen. Die Daten zur Anzeige werden ausschließlich aus Transferobjekten gelesen. Die gesamte Logik der Anwendung wird dagegen in den Klassen des *Business-Layers* gekapselt. Alle Eingaben werden direkt hier adressiert und damit die Steuerung der Applikation vorgenommen.

Keine Methode des *Business-Layer* nimmt einen direkten Datenbankzugriff vor. Die Datenbank wird durch den *Database-Layer* mit einer Persistenzschicht belegt. Hier sind Methoden zur Herstellung und Aufrechterhaltung der Datenbankverbindung und zur Bereinigung der Datenbank hinterlegt. Das Datenbankmodell wird mit einem *Business-Object-Model (BOM)* gekapselt und die Objekte des *BOM* als Transferobjekte an die höheren Schichten übergeben.

Diese Architektur soll einen klar strukturierten Code begünstigen. Dies geschieht gerade vor dem Hintergrund des Einsatzes von PHP als Programmiersprache. Durch dessen Eigenschaft als Skriptsprache geschieht schnell eine Durchmischung von darstellendem Code (HTML-Ausgaben) und Logik (PHP-Anweisungen). Der Code droht durch die häufigen Datenbankzugriffe (SQL-Statements) weiter an Übersichtlichkeit zu verlieren. Die Architektur nach MVC und die konsequente Verwendung von OOP auch unter PHP soll diese Gefahr entschärfen.

Benutzerverwaltungskonzept

Die Anforderungen machen ein Konzept zur Benutzerverwaltung erforderlich. Die Benutzerverwaltung wird durch folgende Rahmenbedingungen gestaltet.

1. Die Benutzerverwaltung bietet die Option zur Registrierung. Diese wird durch einen Screen **`register`** gewährleistet. Es werden die Daten Vorname, Nachname, Emailadresse und ein Paßwort erfasst. Weitere Daten sollen erst in späteren Releases berücksichtigt werden.
2. Das Paßwort wird verdeckt (Sternchen, etc) und zur Sicherheit redundant erfasst und vor der Eintragung auf Identität überprüft.
3. Die Identifikation der Benutzer erfolgt ausschließlich durch die persönliche Emailadresse. Es wird kein separater Benutzername geführt.
4. Aus Gründen des Datenschutzes sind registrierte E-Mailadressen für Dritte niemals sichtbar.
5. Aufgrund der Bedeutung der Emailadresse zur Identifikation und zur Information über anstehende Abstimmungen wird eine Registrierung durch einen Probeversand die Emailadresse verifiziert[131].

131 Ein entsprechendes Flag ist im Datenbankschema berücksichtigt, die Funktion in der ersten Version der Software aber noch nicht umgesetzt.

6. Jeder Benutzer verfügt über ein persönliches Adressbuch. In das Adressbuch können über eine Maske **`address`** Kontakte eingetragen werden. Einträge können zu Gruppen zusammengefasst werden. Dies ist rekursiv möglich. Die Einträge des Adressbuches stehen neben der direkten Eingabe von E-Mmailadressen bei der Einladung zu einer Abstimmung zur Verfügung.
7. Die Maske **`invite`** zeigt das individuelle Adressbuch und ein Feld zur Eingabe weiterer E-Mailadressen. Die Angabe dieser E-Mailadressen sorgt für die Einladung von nicht als Kontakt hinterlegten Benutzer.

Project-Management

Technik

Als Technik kommt auf der Serverseite PHP zum Einsatz. Diese Entscheidung wird vor dem Hintergrund der individuellen Programmierkenntnisse des Projektmitgliedes getroffen. PHP ist frei verfügbar und in allen wichtigen Serverlandschaften schnell und problemlos einsetzbar. Durch die Einführung von Objektorientierter Programmierung ist PHP eine geeignete Alternative auch für mittlere bis große Projekte geworden.

Für die Clientseite wird grundsätzlich ein *XHTML*-Dokument erzeugt. Die Formatierung der angezeigten Elemente erfolgt mittels *CSS*. Als eine grundlegende Regel gilt, dass *JavaScript* für die Betrachtung und Bearbeitung der Seiten des Projektes nicht notwendig ist. Zur optischen Untermalung, für *Highlighting*-Effekte und ähnliches soll *JavaScript* dagegen zum Einsatz kommen. Einem Benutzer sollen also alle Funktionen der Anwendung auch dann uneingeschränkt zur Verfügung stehen, wenn er sämtliche Skriptfunktionen in seinem Browser deaktiviert hat. Dies ist dem Prinzip der Allgemeinheit einer Abstimmung geschuldet, welches zwar grundsätzlich durch die Verwendung eines webbasierten Systems bereits eingeschränkt ist, darüber hinaus aber mit möglichst wenigen weiteren Einschränkungen an die Ausführungsumgebung gestaltet sein soll.

Configuration-Management

Versionierung

Aufgrund der kleinen Projektgröße und aufgrund einiger Überlegungen in der Planung der Projektumgebung soll **kein** Versionierungswerkzeug zum Einsatz kommen. Dies ist für jedes Projekt mit mehr als einem Teammitglied nach modernern Maßstäben eine risikoreiche Entscheidung. Hier ist die Entscheidung vertretbar, da die Hauptaufgabe der Versionierung – das Konfliktmanagement – entfällt. Die zweite wichtige Funktion eines Versionierungswerkzeugs, die Wiederherstellbarkeit alter Konfigurationen muss jedoch mit geeigneten Mitteln nachgebildet werden. Dazu soll eine skriptgesteuerte *Backup*-Funktion zum Einsatz kommen.

Die Disziplin Configuration-Management wird im Folgenden nicht weiter beachtet, auch wenn sie ihre Arbeit während der gesamten Projektdauer

kontinuierlich leistet.

Test

Testfallkarte

Die Disziplin Test stützt sich grundsätzlich auf die Methoden und Bezeichnungen, wie sie im Standard des International Software Test Qualification Board (ISTQB) festgehalten sind.

In einem großen Projekt sollte eine Software zur Verwaltung von Testfällen eingesetzt werden, weil durch die anpassbare (und dann einheitliche) Struktur einer Testfallkarte eine homogene Dokumentation der Testfälle sichergestellt ist. Damit kann das Projekt auch den Wechsel von Mitgliedern der Testmannschaft verkraften. Im Projekt ***rogatio*.net** wird ein Testfall durch einfachen Fließtext beschrieben, der aus vier Teilbereichen besteht.

1. Testdaten
2. Vorbedingungen
3. Testschritte
4. Nachbedingungen

Die Testdaten umfassen Daten, die im Testfall Verwendung finden, also beispielsweise der Titel einer Abstimmung. Auch weitere Daten wie die Nummern der abgedeckten Anforderungen werden geführt.

Vorbedingungen sind Voraussetzungen, um den Testfall ausführen zu können. Ein Beispiel ist die Präsenz eines registrierten Benutzers im System.

Die Testschritte sind die tatsächlich auszuführenden Tätigkeiten, welche so präzise beschrieben werden, dass sie in jeder Testausführung gleich ausgeführt werden können. Es sollte möglichst wenige Spielräume geben.

Die Nachbedingungen müssen vom Tester nach der Testausführung überprüft werden. Greifbar ist die Überprüfung der Datenbank nachdem ein Testfall die Anlage von Daten am System überprüft hat.

Die Testfallkarten sind im Anhang aufgeführt.

Environment

Die Disziplin *Environment* kommt häufig später zum Einsatz. Aber da die Durchführung des Projektes und zudem die einzusetzende Technik zum jetzigen Zeitpunkt bereits feststeht, können geeignete Werkzeuge ausgewählt und aufgesetzt werden.

Entwicklungsumgebung

Als **Entwicklungsumgebung** wird Eclipse zum Einsatz kommen. Für diesen Zweck existieren zwei verschiedene für PHP optimierte Umgebungen. Zum einen das *PHPEclipse*-Projekt [O17] und zum anderen die *PHP Developer Tools (PDT)*. Erstgenanntes Projekt scheint länger zu existieren und hat ein paar Fähigkeiten

mehr, wie beispielsweise Source-Code-Verfolgung auf Dateiebene. *PDT* dagegen wird von den Firmen Zend und IBM unterstützt, Zend steuert einen kostenlosen Debugger bei. *PDT* setzt auf die *Eclipse Web Standard Tools (WST)* auf und bietet damit implizit Unterstützung für Webdateien und XML. Ein Ziel des Projektes ist die möglichst nahtlose Integration in eine bestehende Eclipse-Installation.

Modelliert wird mit dem recht neuen *Eclipse Modeling Framework (EMF)*. Da *WST* unter anderem darauf aufsetzt, ist eine Kombination von *EMF* mit *PDT* möglich. Insgesamt entsteht damit eine mächtige Entwicklungsumgebung inklusive Syntaxcheck, Code-Completion, Syntaxhighlighting und außerdem einer integrierten Modellierungsumgebung. Probleme bereiten verschiedene Versionen und Abhängigkeiten zwischen den *Eclipse*-Derivaten. Insbesondere setzt das *EMF* Eclipse in der Version 3.4 (Ganymed) voraus. Dies schließt zum aktuellen Stand *PHPEclipse* als Alternative aus, weil dies Projekt zum jetzigen Zeitpunkt noch auf Eclipse Europa (Version 3.3) basiert.

Die folgende Übersicht listet die eingesetzten Eclipse-Komponenten und insbesondere deren Versionen.

- Eclipse Modeling Framework (EMF)
  Version: 2.4.1
  Dateiname: emf-runtime-2.4.1.zip
- Graphical Editing Framework (GEF)
  Version 3.4.1
  Datename: GEF-ALL-3.4.1.zip
- Web Tools Platform (WTP)
  Version: 3.0.2
  Dateiname: wtp-R-3.0.2-20080921203356.zip
- Data Tools Platform (DTP)
  Version: 1.6.1
  Dateiname: dtp_1.6.1.zip
- PHP Development Tools
  Version 1.0.3
  Dateiname: pdt-runtime-1.0.3.zip
- Eclipse Classic
  Version: 3.4.1
  Dateiname: eclipse-SDK-3.4.1-win32.zip
- PDT ALL IN ONE
  Version: 1.0.3
  Dateiname: pdt-1.0.3.R20080603_debugger-5.2.14.v20080602-all-in-one-win32.zip
- Equinox (required by PDT)
  Version: 3.4
  Dateiname: eclipse-equinox-3.4.zip

Ein Produkt namens *Webserver On Stick*[132] ist die produktive **Testumgebung**. Das Produkt stellt einen vollwertigen Apache-Webserver samt PHP- und MySQL-Integration bereit. Das besondere: Die Umgebung ist portabel und kann von einem beliebigen Laufwerk aus gestartet werden.

Für die **Backup**-Funktion kommt der freie Packer *7-Zip* [O20] und für die **Dokumentation des Source-Codes** *Doxygen* [O21] zum Einsatz. Textverarbeitung, Tabellenkalkulation und Zeichnungen liefert das *OpenOffice*-Projekt.

Weitere Werkzeuge sind zum jetztigen Zeitpunkt nicht vorgesehen. Anforderungsverwaltung, Planung und Dokumentation erfolgen herkömmlich mit den Mitteln von *OpenOffice*. Dies ist dem Aufwand und dem Charakter als Ein-Mann-Projekt geschuldet. Falls die Zeit bleibt könnte später ein automatisiertes Testwerkzeug wie *Selenium* eingesetzt werden.

Die gesamte Entwicklungsumgebung hat damit die Eigenschaft portable Software zu sein, also unabhängig von einer Installation zu starten. Dies vereinfacht die Auslieferung.

Change-Management

Nach den ersten Veränderungen der Anforderungen in Bezug auf die Zugangszeit schon in der *Inception-Phase*, wird dieser kritische Parameter auch während der Entwurfszeit hinterfragt. Die Disziplin *Business-Modeling* hat ein Zustandsdiagramm erstellt, welches die sechs Zustände charakterisiert, in denen sich eine Abstimmung befinden kann. Danach ist es als Folge von CR01 nicht mehr möglich, eine feste Startzeit vorzugeben, zu der das System dann die Abstimmung automatisch aktiviert. Nun startet eine Abstimmung entweder direkt bei der Erstellung oder auf Knopfdruck des Administrators. Beendet wird sie entweder nach Ablauf einer optional eingestellten Dauer, durch Abbruch ausgelöst von Administator oder Teilnehmer oder durch Abgabe aller möglichen Stimmen. Ein Teil dieser Vorgaben waren in der Anforderung mit der Nummer REQ007 beschrieben, allerdings nicht vollständig. Der Change Request CR02 präzisiert dies zur Sicherstellung der Ergebnisse.

Weiterhin hat die Disziplin *Analyses&Design* eine Lösung gefunden, Reports passend zu verteilen ohne zeitgesteuerte Ereignisse zu verwenden. Danach werden im Moment des Endes einer Wahl keine Benachrichtungen versendet, sondern beim Besuch der Seite der Report angezeigt. Der Administrator hat die Möglichkeit die Versendung des Reports gesondert anzustoßen. Die Vermeidung von Cron-Jobs muss in Anforderung REQ009 nachgezogen werden, dies wird mit CR03 wie oben beschrieben erledigt. Im Gegensatz zum Start einer Abstimmung erscheint der verzögerte und optionale Versand von Benachrichtungen zum Ende verschmerzbar, möglicherweise sogar wünschenswert, um das System nicht als zu geschwätzig erscheinen zu lassen. Andere Optionen wird die Disziplin *Requirements* als Anforderung der Priorität „3-NICE“ führen.

132 Die Software wurde vom Entwickler zwischenzeitlich unter dem Namen Movable-Web-Server (MoWeS) als Open-Source freigegeben, [O19].

2.2 Iteration E2

Implementation

CSS-Boxmodel

Die Disziplin Implementation hat einen Prototyp für die graphische Darstellung erarbeitet, die sogenannte Simple-Module-Library (SML). Sie setzt das CSS-Boxmodell in eine PHP-Schnittstelle um. Demnach besteht die Oberfläche der

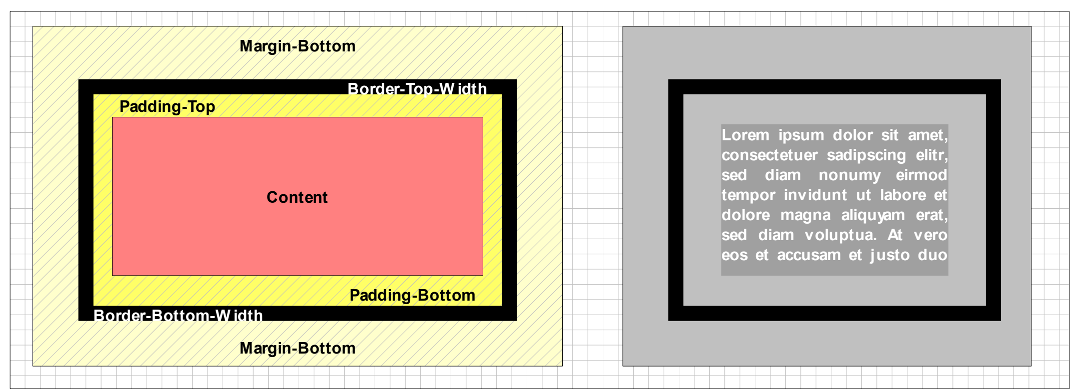


Abbildung 5: Das Boxmodell der Cascading-Style-Sheets

Webanwendung aus Modulen, die eine feste Größe in Pixel erhalten.

PHP-Konstruktor

Wird in PHP (mindestens bis zur aktuellen Version 5.3) eine Klasse B von einer Klasse A abgeleitet, so findet bei oder vor dem Aufruf des Konstruktors der Klasse B kein impliziter Aufruf des Konstruktors der Klasse A statt. Aus Gründen der Abwärtskompatibilität ist auch nicht zu erwarten, dass sich dieses Verhalten einmal ändern wird. Dies bedeutet, dass nicht sicher davon ausgegangen werden kann, dass notwendige Initialisierungen von privaten Attributen eines Objektes vorgenommen werden, wenn die betroffene Klasse abgeleitet wird. Aufgrund ihres Charakters als Bibliothek ist eine sinnvolle Vererbung der der Klassen der Simple Module Library vorstellbar, so dass die Deklaration als nicht vererbbare (finale) Klasse keine Option darstellt. Weiterhin ist es in PHP weder durch die Instantiierung neuer Objekte noch über einen Methodenaufruf möglich eine Wertzuweisung an Objektvariable vorzunehmen.

Die Lösung des Problems der Initialisierung stellt ein Entwurfsmuster namens „Lazy Instantiation“ dar. Es wird auch bei der Verwendung des Musters „Singleton“ verwendet, darf aber nicht mit ihm verwechselt werden. Lazy Instantiation bedeutet, dass eine Variable erst bei der ersten Verwendung, also beim ersten Aufruf des zugehörigen Getters angelegt wird. Dies hat zudem den Vorteil, dass nicht benötigte Attribute auch keine Systemresourcen belegen.

Als Konsequenz sind alle Attribute der Klassen der SML privat und nur über Getter und Setter zugänglich. Sie werden grundsätzlich mit dem leeren Zeiger

(null) deklariert. Eine initialisierende Wertzuweisgung erfolgt immer durch Lazy Instantiation, zur Klarheit auch im Falle von primitiven Datentypen wie Zeichenketten. Davon ausgenommen sind interne Variable, die beispielsweise zur Verwaltung eines internen Status dienen und keine nach außen sichtbaren Attribute darstellen. Hier würde es keinen Sinn machen künstlich private Getter und Setter zu erzeugen.

3. Construction Phase

Die Construction-Phase umfasst viele Aspekte der Entstehung der Software. Sie können nicht alle erwähnt werden. Es wurden drei Iterationen durchgeführt. Die ersten Iteration beschäftigt sich mit „normaler“ Implementierungsarbeit, die zweite Iteration diente dem aufgrund der Softwaregröße notwendigen Refactoring und die dritte Iteration hatte eine dem Zeitdruck geschuldete Funktionsreduzierung und einen öffentlichen „Beta“-Test[133] zum Inhalt.

3.1 Iteration C1

Analysis & Design

Modulprinzip

Ein besonderes Problem stellt für die geplante Webanwendung die Oberflächenprogrammierung dar. Viele bekannte PHP-Bibliotheken für diesen Zweck basieren auf *JavaScript* oder *AJAX* und kommen aufgrund der Projektphilosophie nicht in Frage. Für ***rogatio*.net** soll daher eine eigene Oberflächendarstellung implementiert werden. Die Bibliothek hat zunächst nicht den Anspruch generisch für beliebige Webanwendungen einsetzbar zu sein. Prinzipiell sollen die Schnittstellen aber für einen solchen Einsatz erweiterbar bleiben. In dieser Phase wird zur Evaluation ein *Exploratory Prototype* entworfen.

Das GUI-Framework basiert auf einem Ansatz des Designers Andy Clarke[134]. Demnach ist die Gestaltung jeder Webseite auf ein Gitterlayout zurückzuführen. Diese Idee wird zum Prinzip erhoben. Mit dem Prototyp sollen drei Aspekte dieses Prinzips evaluiert werden:

1. **Modul**. Die Abbildung des Begriffes des GUI-Moduls (Feld eines Gitters) auf das CSS-Boxmodell und die Kapselung dieser Abbildung in eine PHP-Klasse.

2. **Layout**. Ein Modul kann weitere Module als Kindelemente enthalten. Diese sollen nach einem definierbaren Layoutalgorithmus angeordnet werden.

3. **Positionierung.** Die Positionierung der Module erfolgt im Modus Offsetting (CSS-Property `position:relative`), das heißt per CSS werden die Startkoordinaten der linken oberen Ecke eines Moduls relativ zum Elternelement vergeben. Es sollen die beiden Möglichkeiten einer festen Größe und eines sogenannten elastischen Layouts[135] evaluiert

133 Die Softwareversion hatte zu diesem Zeitpunkt Alpha-Status, mit „Beta“-Test ist eine Benutzerakzeptanztest gemeint.

134 Siehe Clarke/Holzschlag, Transcending CSS – Neue kreative Spielräume im Webdesign, S. 183ff.

135 Elastische Layouts entstehen durch die Verwendung der Einheit em. Sie können im Browser

werden.

Modul

Ein Modul wird abgebildet durch ein DIV-Element. Eine PHP-Klasse trägt die Eigenschaften des CSS-Boxmodells. Durch die Formatierung per CSS werden der PHP-Klasse Module weitere Eigenschaften hinzugefügt, die das Design betreffen, also Farbe, Schriftart und Rahmengestaltung.

```
<?php
abstract class StyleModule extends Showable
{
 private $background_color = "#DDDDDD";
 private $background_image = null;
 private $background_repeat = "no-repeat";
 private $border_top_color = "#000000";
 private $border_top_style = "solid";
 private $border_top_width = "1px";
 private $color = "#FF0000";
 private $width = "300px";
 private $height = "200px";
 private $margin_top = "0px";
 private $padding_top = "0px";
 private $top_position = "0px";
 private $left_position = "0px";
 private $font_family = "sans-serif";
      private $font_size = "12pt";

      public function set_alignment_center()
      {
            $this->text_align = "center";
      }

      public function get_alignment()
      {
            return $this->text_align;
      }
}
?>
```

Der Quelltext zeigt einen auszug aus den Eigenschaften. Alle Eigenschaften sind durch Getter und Setter geschützt. Die Nomenklatur ist an CSS angelegt. Die Art der Setter verhindert falsche Parameter, also statt

```
public function set_alignment(„left“ | „right“ | „center“)
```

gibt es die Methoden

```
public function set_alignment_left();
public function set_alignment_right();
public function set_alignment_center();
```

einem Zoomfaktor unterworfen werden. Siehe Wikipedia unter dem Stichwort Em_(Schriftsatz). Beispiele für elastisches Layout finden sich im CSS Zen Garden [O22].

Layout

Jedes Modul verwaltet eine Liste von Kindmodulen (**`$modules`**), die auch leer sein kann. Die Kindmodule werden durch die Angabe der CSS-Eigenschaften **`top`** und **`left`** innerhalb des Elternelementes ausgerichtet. Dies leistet CSS und die Interpretation durch den Browser automatisch. Um das manuelle Setzen von gleichverteilten Spalten einer Tabelle zu erleichern, wird vor der Anzeige der Module ein Layoutalgorithmus (**`show_layout`**) gestartet. Mit PHP ist eine Art funktionaler Programmierung möglich, indem der Name einer Funktion in einer Variable (**`$this->layout`**) gespeichert und übergeben werden kann. Dies erlaubt das elegante setzen (Zeile 9) und ausführen (Zeile 20) der passenden Layoutfunktion. Der folgende Quelltext zeigt das Prinzip:

```
<?php
abstract class LayoutModule extends ScriptModule
{
 private $modules = array ();
 private $layout = null;

 public function set_horizontal_grid_layout()
 {
 $this->layout = "do_horizontal_grid_layout";
 }

 public function show_layout()
 {
 // buffer function pointer
 $local = $this->layout;

            // call layouting function
            if ($local != null)
            {
                  $this-> $local ();
            }

            foreach ($this->modules as $m)
            {
                  $m->show();
            }
      }
}
?>
```

Die aufgerufene Layoutfunktion ordnet nun die Kindelemente relativ zum Elternelement an:

```
private function do_horizontal_grid_layout()
{
      $c = count($this->modules);
      if ($c > 0)
      {
            $width = floor($this->getWidth() / $c);
            for ($i = 0; $i < $c; $i++)
            {
                  // calculate
                  $m = $this->modules[$i];
```

```
                $p = $width * $i;
                $h = $this->getHeight() - $m->getOuterHeight();
                $w = $width - $m->getOuterWidth();

                // set values
                $m->setHeight($h);
                $m->setLeftPosition($p);
                $m->setTopPosition(0);
                $m->setWidth($w);
            }
        }
}
```

Positionierung

Die Idee eines elastischen Layouts ist vor dem Hintergrund von barrierefreiem Web eine optimale Lösung. Die Browser bieten die Möglichkeit einen Zoomfaktor zu setzen und dadurch seebehinderten Menschen einen besseren Zugang. Der Prototyp zeigt aber, dass die Unterstützung der Browser aktuell noch nicht gegeben ist. Der noch immer verbreite Internet-Explorer 6 hat allgemeine Schwierigkeiten im Layout (im Screenshot wird die Schrift nicht zentriert). Probleme ergeben sich insbesondere mit Hintergrundbildern. Diese werden erst mit Mozilla 3 in der Größe angepaßt. zeigt den Unterschied. Aufgrund der ohnehin schwierigeren Berechnungen wird das Projekt daher kein elastisches Layout verwenden.

Ergebnis

Das Ergebnis des *Prototyping* ist positiv. Das Konzept des fixen Layouts hat sich bewährt, es ist leicht in einem Objektmodell kapselbar. Das Layout ist hochflexibel und trotzdem serverseitig leicht zu handhaben. Abbildung 6 zeigt die Optionen des Layoutmanagers der Simple-Module-Library. Durch die Verschachtelung der Module lassen sich aus den dargestellten Basis-Layouts komplexe Kombinationen erstellen.

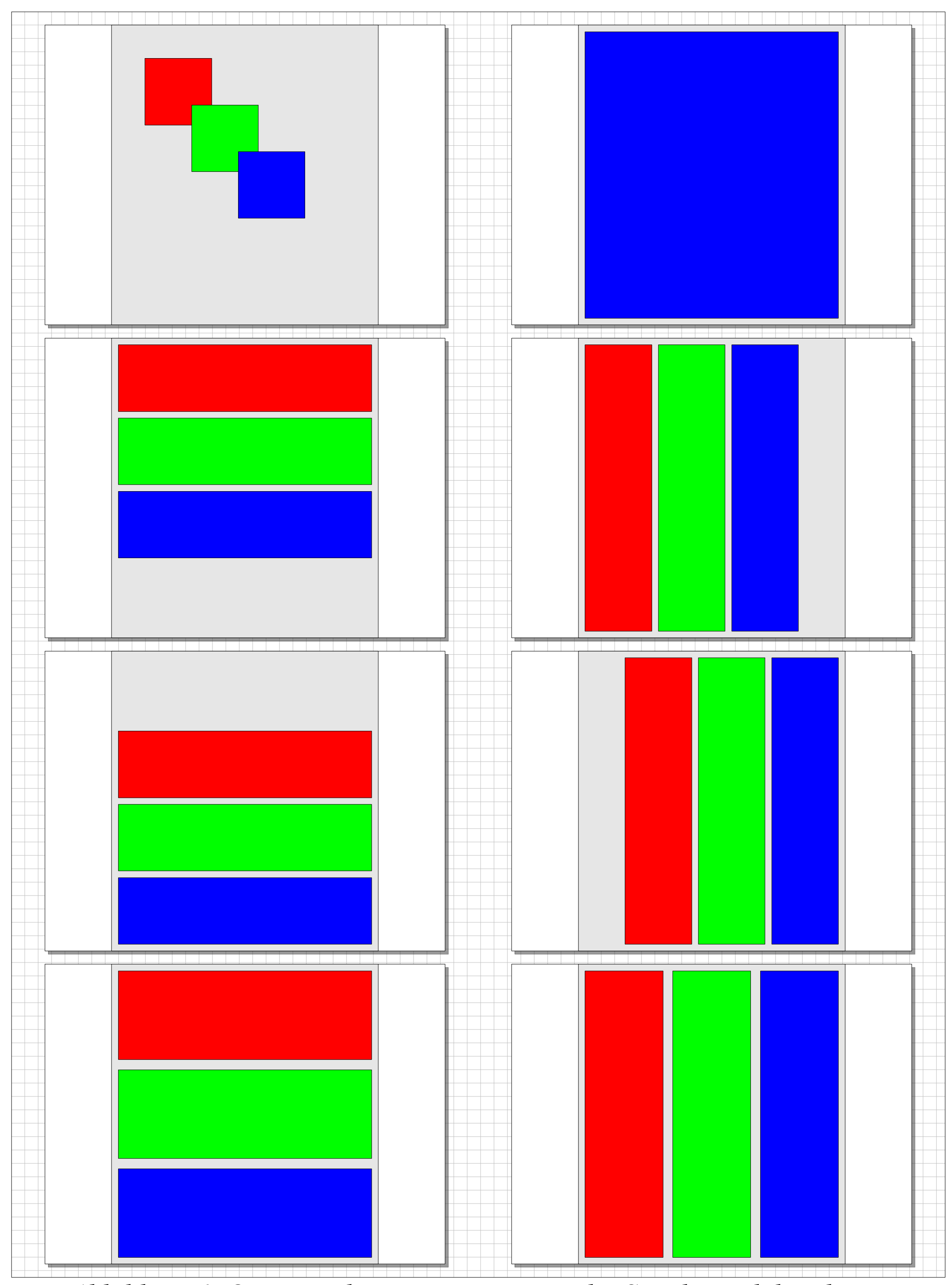


Abbildung 6: Optionen des Layoutmanagers der Simple Module Library

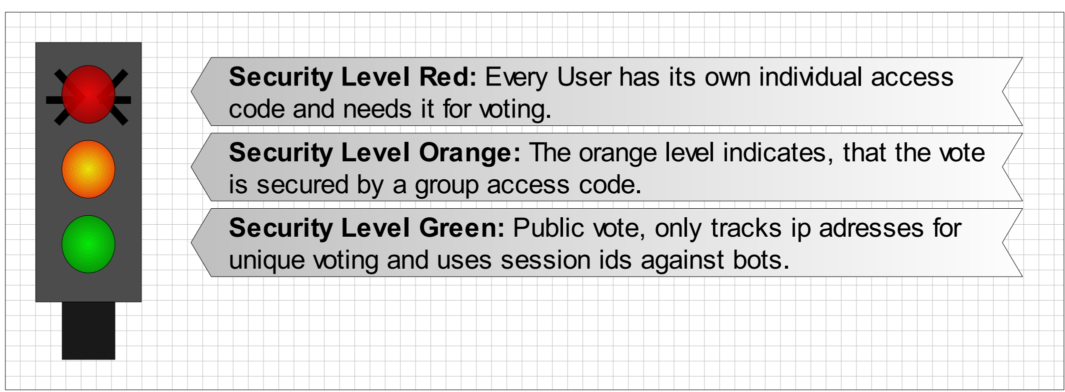


Abbildung 7: Sicherheitslevel

Zugangssicherheit

Bei einer Anwendung zur Erfassung von Wählerstimmen stehen Sicherheitsaspekte im Vordergrund, dies ist umso mehr der Fall, wenn es sich um eine Webanwendung handelt. Gleichzeitig drohen einer großen Menge von verbreiteten Webanwendungen die gleichen Gefahren und daher sind die Konzepte zum Schutz ausgefeilt.

Angriffe auf die Transportschicht

Die Kommunikation zwischen Webserver und Webclient erfolgt über das Protokoll HTTP, welches immer auf eine zuverlässige Transportschicht aufsetzt, in nahazu allen Fällen auf TCP. Alle Angriffe, die in der Transportschicht stattfinden sind für die Webanwendung nicht kontrollierbar. Das Sicherheitskonzept des Projektes geht daher grundsätzlich davon aus, dass die Transportschicht gesichert ist und kein Dritter den Datenverkehr mitlesen kann, auch nicht durch durch einen Sniffer im Netzwerk oder gar Funknetzwerk. Damit scheiden Janusangriffe[136] bei der Betrachtung aus. Die Anforderung kann durch Installation der Anwendung auf einem Webserver, der das HTTPS-Protokoll unterstützt sichergestellt werden, die sichere Übertragung von Emails im Funknetzwerk durch POP3S[137].

Drei Sicherheitslevel

Das Projekt Forum Publicum setzt drei denkbare Szenarien für E-Voting um, die Wahl innerhalb eines festen wahlberechtigen Personenkreises, der im Vorfeld der Wahl bekannt ist und eine etwas flexiblere Variante, die mit Einladungen an Emailadressen arbeitet. Abbildung 7 zeigt eine Übersicht mit den wesentlichen Rahmenbedingungen. Details zur Benutzung der Level finden sich im Online-Handbuch und sollen an dieser Stelle nicht wiederholt werden.

136 Auch bekannt als Man-In-The-Middle-Attack.

137 POP3S ist ein eine Erweiterung von POP3 um eine Verschlüsselung durch SSL. Dadurch wird nicht nur das Paßwort nicht mehr im Klartext übertragen, auch das Mitlesen von Emails ist erst durch Brechen der SSL-Verschlüsselung möglich.

Datenbankschema

Die Persistenzschicht muss das Laden, Speichern und Aktualisieren der Objekte des Datenmodells gewährleisten. Dabei entsprechen die Objekte des Datenmodells den Tabellen der Datenbank. Referenzen werden durch flache Abbildungen in automatisch inkrementierten Identifikationsnummern aufgelöst. Prinzipiell sind dabei drei Multiplizitätenkategorien möglich: „1:1“, „1:n“ und „n:n“.

In der Beziehung „1:1“ gehört zu jedem Objekt genau ein weiteres Objekt oder dieses Objekt ist undefiniert[138]. Das Datenmodell von ***rogatio*.net** ist klein genug, um solche Abbildungen nicht vorkommen zu lassen. Denkbar wäre die Ausgliederung von Start- und Endzeiten in eine eigene Klasse und in eine eigene Tabelle. Dies ist in der ersten Version nicht vorgesehen. Vorgehen ist in der Version aber die Vereinfachung, dass zu jeder Wahl genau ein Optionsobjekt gehört, in welchem mögliche Antworten („Ja“, „Nein“, „Enthaltung“) gespeichert werden. Erst zukünftige Versionen werden dies flexibilisieren. Damit ist die Abbildung Vote nach Option aktuell eine „1:1“-Beziehung.

Die Beziehung „1:n“ ordnet einem Objekt kein Objekt, genau ein Objekt oder viele weitere Objekte zu. Auf Klassenebene geschieht dies durch eine Liste von Referenzen, auf Datenbankebene durch die Speicherung der Besitzeridentifikation in der Zieltabelle. Ein Beispiel ist das Verhältnis Wahl zu Kommentar. Eine Wahl hat viele Kommentare. Die Klassen Vote hat eine Liste von Referenzen auf die Klasse Comment. Die Tabelle comments hat eine Spalte vote_id mit Identifikationsnummern der Tabelle votes.

Komplizierter sind die „n:n“-Beziehungen. Die erste Version von ***rogatio*.net** enthält mit dem Adressbuch ein Beispiel. Jeder Benutzer kann jeden anderen Benutzer in seinem Adressbuch aufnehmen. Auf Klassenebene hat jedes Objekt vom Typ User eine Liste von Referenzen auf seine Adressbucheinträge. Auf Datenbankebene existiert eine spezielle Verlinkungstabelle mapping, welche einer Identifikation admin_id eine einzelne user_id zuordnet. Jede einzelne Verbindung zwischen Benutzern wird so flach abgebildet[139].

Speichern: Alle drei Verhältnistypen („1:1“, „1:n“, „n:n“) muss durch die Persistenzschicht abgebildet werden. Zur Speicherung existiert die Klasse Saver. Zunächst benötigt jede Datenmodellklasse eine Funktion zur direkten Persistierung in der zugehörigen Datentabelle. Dazu gibt es die Einfügemethoden (Insert). Zum Einfügen einer Wahl speichert die Methode

```
public static function insert_vote($vote)
```

die Wahl in die Tabelle votes. Dabei wird eine Identifikationsnummer generiert, dem Objekt zugewiesen und zurückgeliefert.

Zum Einfügen von abhängigen Objekten wie einem Kommentar, gibt es Methoden mit erweiterter Signatur. Die Methode

138 Technisch gesprochen existieren weitere Kategorien, das Verhältnis „1:1“ würde in einer Modellierung in UML durch „1:1“ und „1:0“ abgebildet.

139 Das Beispiel stammt aus ersten Überlegungen. Mit der Einführung der benutzerdefinierten Kontakten entfallen n:n-Beziehungen aus dem Projekt.

```
public static function push_comment($vote_id, $comment)
```

persistiert einen Kommentar. Damit wird „1:1" realisiert. „1:n" geschieht durch wiederholten Aufruf der Methode.

Zur Abbildung von der „n:n"-Beziehung zwischen den Benutzern im Adressbuch gibt es die Methode

```
public static function link_user_users($user)
```

Die Beziehungen im Objektmodell werden durch Referenzen des Objektes user übergeben und müssen von der Methode selbstständig aufgelöst werden.

Die Signaturen spiegeln den Einsatzzweck der Persistenzschicht in einem skriptbasierten Webumfeld wieder. Es gibt bei der Anfrage an die Webapplikation keinen Objektlebenszyklus, der aus Laden, wiederholter Manipulation und Speichern mit Schließen der Applikation besteht. Daher wird in einem Durchlauf der Software immer nur eine Aktion durchgeführt. In der Anzeige einer Wahl wird ein Kommentar hinzugefügt, dabei die Wahl aus der Datenbank geladen, das

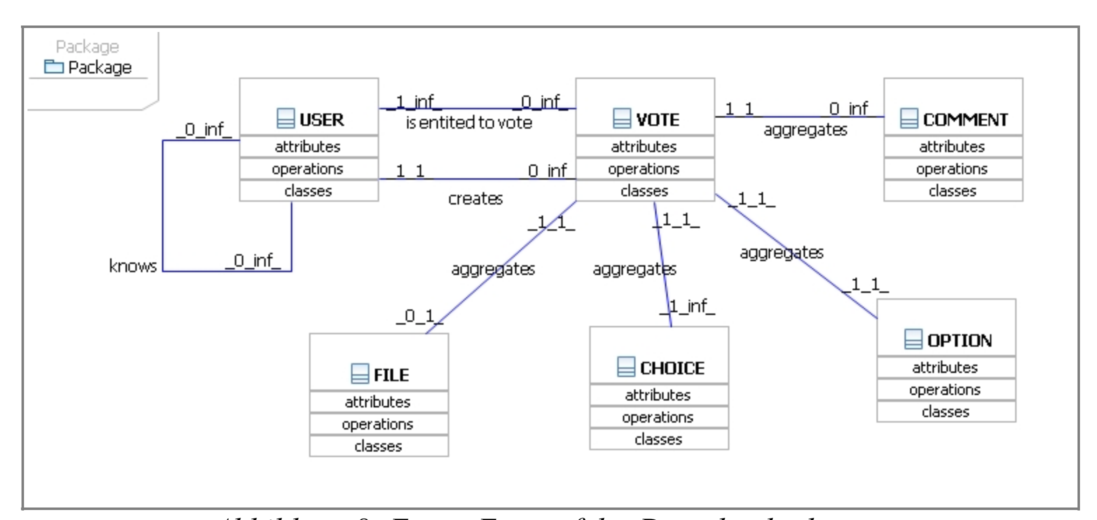


Abbildung 8: Erster Entwurf des Datenbankschemas

Objekt Comment generiert und zum Abschluß persistiert. Daher ist die Modellierung als push_comment($vote_id, $comment) zur Speicherung eines Kommentars sinnvoller als push_comments($vote) mit eigener Auflösung der Referenzabhängigkeiten. Denn diese müßten von der aufrufenden Seite zunächst ebenfalls zusätzlich aufgebaut werden.

Bei der Persistierung von „n:n"-Beziehungen in den „Link"-Methoden werden daher auch nur teilweise befüllte Referenzen benutzt. In link_vote_users($vote) enthalten die referenzierten Benutzer der Klasse User keine vollständigen Daten, sondern die vom Transferprotokoll übertragenenen Identifikationsnummern der im Adressbuch ausgewählten Benutzer.

Implementation

Cloning

Mit der Objektorientierung wurde in PHP auch ein Konzept zum Klonen von Objekten eingeführt. Das Schlüsselwort `clone` erzeugt eine Kopie eines Objektes und kopiert sämtliche Attribute konsequent. Dies bedeutet, dass Zeigertypen die Adresse des Originalzeigers erhalten und damit die Zieladresse des Originals. Da Objekttypen in PHP implizit Referenzen sind, werden diese nicht mitgeklont. Ein generelles „Deep Cloning“ wäre auch nicht wünschenswert, weil es im Falle von zirkularen Referenzen zu einer Endlosschleife führen würde. Felder werden elementweise nach dem beschriebenen Muster kopiert, bei Feldern mit Objektreferenzen gelten daher ebenfalls die vorgenannten Überlegungen.

Die ausgezeichnete Methode `__clone` wird zusätzlich auf allen Objekten ausgeführt deren Klassen diese implementieren. Der Aufruf erfolgt auf der Kopie und zwar im Anschluss an den Kopiervorgang. Der Programmierer kann in dieser Methode Werte korrigieren, die nicht exakt kopiert werden sollen. Beispiele sind eine (eindeutige) Identifikationsnummer oder ein Hash-Wert. Die Methode `__clone` kann auch verwendet werden, um gezieltes Deep-Cloning manuell zu realisieren, das folgende Codefragement zeigt ein Beispiel dafür aus SML.

```
/**
 * @details The method updates identifier only
 * if it is existing already to prevent determinism.
 * First setting of identifier is performed in getter
 * always. The array of child modules are cloned manually
 * to perform deep cloning.
 */

public function __clone()
{
      if ($this->ident)
      {
            $p = self :: $prefix;
            $n = ++self :: $runner;
            $this->ident = $p . "_" . $n;
      }

      for ($i = 0; $i < $this->getModuleCount(); $i++)
      {
            $this->modules[$i] = clone $this->modules[$i];
      }
}
```

Die Methode `__clone` realisiert damit ein anderes Konzept als beispielsweise die Methode Object.clone() der Java-API, sie kann nicht überschrieben werden und auch ein leerer Methodenrumpf ändert nichts an dem Klonsprachmechanismus.

```
public function __clone()
{
	if ($this->ident)
	{
		$p = self :: $prefix;
		$n = ++self :: $runner;
		$this->ident = $p . "_" . $n;
	}
}
```

Test

Das Projekt ***rogatio*.net** benötigt in seiner Aufstellung als Ein-Mann-Projekt kein ausgefeiltes Testkonzept. Grundlegende Basis für den Aufbau von Teststufen ist das V-Modell, wie es auch im ISTQB-Standard Verwendung findet. Aufgrund der Projektgröße können allerdings nicht alle Ebenen vollständig bearbeitet werden. Dies würde auch wenig Mehrwert bieten. Ziel ist vielmehr eine sinnvolle Stichprobe. Diese Vorgehensweise ist eine immanente Eigenschaft des Testmanagements: Vollständige Testabdeckung ist in der Regel unmöglich.

Das Testmanagement stützt sich daher auf vier Säulen zur Sicherstellung der Produktqualität.

1. Entwickler-*Smoke*-Tests
2. Anforderungsbasiertes Testen
3. *Use-Case*-basiertes Testen
4. Offener Feldtest

Das Testmanagement hat im Kern die Aufgabe, die Anforderungen an das System zu verifizieren, also sicherzustellen, dass das System die Fähigkeiten tatsächlich hat, die es haben sollte. Mit dem Test einer einzelnen oder wenigen Anforderungen wird die Ebene der Integrationstests umgesetzt, denn diese Tests können auch schon vor einer Gesamtintegration sichergestellt werden.

Die dem Projekt gegeben Anforderungen beschränken sich auf eine selektive funktionale Sicht, es gibt keine Vorgaben über den konkreten Ablauf. Daher müssen in den Test Erkenntnisse der Disziplin *Analysis&Design* einfließen. Die dort definierten *Use-Cases* müssen in Testfälle umgewandelt werden. Dies wird in der zweiten Säule dieses Testkonzeptes getan. Ein Beispiel für einen zu prüfenden Anwendungsfall wäre die Registrierung eines Benutzers, der anschließenden Abfrage der Bestätigungsemail samt Aktivierungslink, der abschließenden Aktivierung und dem schließlichen Anmelden am System mit dem neuen Benutzernamen. Um einen kompletten *Workflow* durch das System prüfen zu können, ist in der Regel ein vollständiges Release der Software Voraussetzung. Daher wird mit der zweiten Säule des Tests im Projekt die Ebene eines Systemtests aus dem V-Modell umgesetzt.

Integrations- und Systemtests werden in der Regel von einer unabhängigen

Testabteilung oder -mannschaft konzipiert und durchgeführt, um Voreingenommenheiten und Betriebsblindheit zu umgehen. Um grobe Fehler und Fehler auf einer tiefen Systemebene frühzeitig zu erkennen, sollten diesen Testvarianten immer Komponententests vorausgehen. Diese Tests sind in der Regel Entwicklertests, werden also von der Entwicklungsmannschaft konzipiert und durchgeführt. Dies hat seine Ursache in der Notwendigkeit weitreichender Kenntnisse des Code-Designs.

Während der Entwicklung von Komponenten macht es besonders bei einer Webapplikation Sinn, Seiten immer wieder schnell im Browser anzeigen zu lassen. Diese Form von Tests werden Smoke-Tests genannt. Sie werden im Projekt aus Entwicklersicht betrieben, um so einen agilen Ansatz besonders bei der Gestaltung der Oberfläche umsetzen zu können.

Ziel des Projektes ***rogatio*****.net** ist es unter anderem, Benutzer zu Abstimmungen einzuladen, die bisher keine Berührung mit dem System hatten. Daher ist eine sehr gute Zugänglichkeit der Seiten notwendig. Zur Überprüfung dieser weichen Faktoren ist nach Fertigstellung des ersten Release der Software ein Feldtest geplant. Der Feldtest findet in einer produktiven Umgebung statt, in der das System in Zukunft betrieben wird. Der Feldtest hat damit den Charakter eines Beta-Test. Um drohende Betriebsblindheit zu vermeiden, ist das Ziel betriebsfremde Probanden nur unter Kenntnis des Testfalles zu bitten, das System zu bedienen und den Erfolg und deren Feedback zu protokollieren.

Entwickler-Smoke-Tests

Smoke-Tests sind nicht aus systematischen Testüberlegungen entstanden und folgen nicht immer festen Mustern. Im Kern ist es der Antrieb eines jeden Entwicklers nach Fertigstellung einer Komponente oder einer Veränderung das Ergebnis der Arbeit zu betrachten. Viele Entwickler starten während der Arbeit wohl ganz unbewußt diese Art von Test, um zu sehen „was passiert“. Diese Tätigkeit ist selbstverständlich und elementar für die Übergabe eines Releases an die Testmannschaft, weil das Auffinden einer Fehlerwirkung samt Meldung an die Entwicklung für einen Tester aufwendiger ist, als für den Entwickler selbst. Weiterhin ist die Suche nach der Fehlerursache in zeitlicher Nähe zur Programmierung der Komponente einfacher.

Smoke-Tests machen auch vor dem Eintritt in einen Testzyklus von seiten der Testmannschaft Sinn, um überflüssige Aufwände in der Testvorbereitung zu vermeiden. Dies wird in der Literatur auch Sanity-Test genannt. In dem Fall soll die Testbarkeit sichergestellt werden, nicht die Meldung von Fehlerwirklungen steht im Vordergrund.

Die Durchführung von Smoke-Tests kann also unterschiedlich ausführlich formalisiert werden. Im diesem Projekt werden sie während der Entwicklung verwendet, aber nicht konkret beschrieben. Die technische Realisierbarkeit dieser Form von Tests wurde durch die Disziplin *Environment* sichergestellt, die einen Webserver samt Eclipse-Integration bereitgestellt hat. Dieser ist in der Lage Anfragen an die Inhalte des Eclipse-Workspaces zu beantworten und damit die Wirkungen jeder Veränderung am Code direkt anzuzeigen.

Anforderungsbasiertes Testen

In der einzigen Iteration der *Inception-Phase* wurden die Anforderungen an das Projekt aufgestellt. In der Umsetzung einer wesentlichen Eigenschaft des *Rational Unified Process* (iteratives Projektvorgehen, Output wird Input) fließen die Erkenntnisse der Disziplin *Requirements* direkt in die Konzeptionierung von Testfällen ein.

Das anforderungsbasierte Testen untersucht den sogenannten Vorwärtsfall. Es handelt sich also um einen Funktionstest. Die Erfolg bei der Erstellung von Testfällen hängt eng mit der Güte der Tätigkeiten der Disziplin *Requirements* ab. Bei ausgefeilter Anforderungsbasis ist die Umsetzung in Testfälle direkt möglich. Das Hauptaugenmerk liegt in der Auswahl der Abdeckung. Die vorliegende Anforderungsbasis ist redimentär und hat einen starken Fokus auf funktionalen Beschreibungen. Diese Anforderungen sollten durch den Test komplett abgedeckt sein. Die Anforderungen wurden in elf Testfälle sortiert, welche jeweils die folgenden Aspekte abdecken.

1. Generelle Funktionalität (TF01)
2. Teilnehmerauswahl (TF02)
3. Aktivierte Abstimmungen (TF03)
4. Abstimmungsende (TF04)
5. Informationsemailversand (TF05)
6. Abstimmung bearbeiten (TF06)
7. Report (TF07)
8. Grenzwertanalyse (TF08)
9. Abstimmung abbrechen (TF09)
10. Abstimmung klonen (TF10)
11. Zugangsbeschränkung (TF11)
12. Adressbuch (TF12)

Der Testfall TF01 hat neben seiner Entstehung aus Anforderungen, die Aufgabe eines Smoke-Tests. Nach jeder Veränderung am Code, sollte dieser grundlegende Fall der Anlage und Abgabe einer Stimme zu einer Abstimmung immer ohne Fehler durchführbar sein.

3.2 Iteration C2

Analysis & Design

Fehlerkonzept

Trotz der beschränkten Größe des Projektes ist ein zuvor durchdachtes Fehlerkonzept sinnvoll, weil mit wachsener Größe des Projektes wird eine nachträgliche Einführung zunehmend schwieriger. Weiterhin gibt es in der Interaktion mit dem Benutzer immer Ausnahmesituationen, die behandelt werden müssen.

Erstaunlicherweise ist gar nicht offensichtlich klar, was ein „Fehler“ eigentlich ist. Eine Division durch Null sicherlich oder der fehlende Netzzugriff. Aber auch ein Syntaxfehler im Quelltext? Oder die Auswahl von nicht registrierten

Emailadressaten in der nur für registrierte Nutzer zulässigen Sicherheitsstufe? Das Fehlerkonzept beachtet grundsätzlich nur Laufzeitfehler. Syntaxfehler, Systemausfälle oder Serverprobleme werden nicht berücksichtigt. Damit folgt die Modellierung dem Prinzip „Keep it Simple“ und der Tatsache, dass Server- und Systemausfälle durch Fehlerseiten und Syntaxfehler durch das Testmanagement bearbeitet werden. Bei einer klassischen Applikation im Stil des Web 2.0, welche auf der Eingabe von Daten in Hypertextformularen samt Aufnahme in einer Datenbank beruht, ist eine wichtige Aufgabe die Konsitenz, Vollständigkeit und Korrektheit der Eingaben zu überprüfen. Werden die gegebenen Rahmenbedingungen verletzt, so wird dies im Sinne dieses Fehlerkonzeptes nicht als Fehler betrachtet. Nachfragen und die klassischen Hinweise auf fehlende Eingaben („Bitte bestätigen Sie das Lesen der Lizenzbedingungen“) sind vielmehr Abweichungen vom schnellsten Weg zum Ziel durch den definierten *Workflow*. Es ist kein Fehler, notwendige Eingaben erst nach einem Hinweis zu tätigen sondern muss in einer sauberen Modellierung durch den *Workflow* abgedeckt sein. Um diese Abweichungen zu behandeln und den Benutzer sauber durch den Workflow zu führen ist auch hierfür ein Konzept erforderlich. Das Fehlerkonzept unterscheidet also in der Behandlung zwischen drei Bereichen.

1. **Kompilierfehler**. Damit sind die Fehler gemeint, die nicht zur Laufzeit oder während der Benutzung der Anwendung auftreten. Darunter fallen Syntaxfehler, Serverausfälle, die den Zugriff auf die Programmdateien verhindern, fehlende Dateien und Konfigurationsfehler, welche die korrekte Interpretation des Codes verhindern.
2. **Umwege**. Damit sind Abweichungen vom direkten Pfad zum Ziel gemeint. Diese Fehler treten besonders bei der Dateneingabe durch den Benutzer auf, aber auch eine fehlende Registrierung fällt in diesem Bereich.
3. **Ausnahmen**. Damit sind die Fehler gemeint, die in höheren Programmiersprachen durch das eben so benannte Exceptionhandling abgefangen werden. Hierunter fallen gescheiterte Zugriffe auf Datenbanken und unerwartete Reaktionen auf Benutzereingaben.

Implementation

Refactoring

Ziel dieser Iteration der *Construction-Phase* ist auch ein *Refactoring* der vorhandenen Quelltexte ohne Veränderung der Funktionalität.

Im Projekt hat es im Laufe der Implementierung Probleme mit der eindeutigen Bezeichnung der Klassen gegeben. Dies liegt im Design der Programmiersprache begründet, in der aktuellen und verwendeten Version 5.2.3 unterstützt PHP noch keine Namensräume, wie C# mit dem Schlüsselwort „namespace“ oder Java mit dem „package“-Konzept[140]. Daher sind gleiche Klassennamen nur eingeschränkt möglich und selten sinnvoll. Weiterhin braucht das zur Dokumentation des Quelltextes eingesetzte Tool *Doxygen* eindeutige Klassennamen. Eine

140 Die Einführung genau dieser Namensräume wird vielfach gefordert (siehe zum Beispiel die „PHP Namespace Petition“ unter http://www.petitiononline.com/phpns/petition.html). Die PHP-Community führt darüber und um die Form der Einführung eine Diskussion.

Möglichkeit, die zunächst genutzt wurde war die Identifikation über Dateinamen, so gab es die „Generic*“-Klassen, also GenericTextfield, GenericRadiobutton und ähnliche, daneben dann „*Input“-Klassen (TitleInput, RequestInput), „*Output“-Klassen (TitleOutput, RequestOutput). Dies alles in einem Ordner für die Klassen der Schicht Presentationlayer, wurde schnell unübersichtlich.

Die Lösung soll ein Designmuster der objektorientierten Programmierung bringen, genannt *Abstract-Factory*. Alle darstellenden Klassen verfügen im Wesentlichen nur über einen Konstruktur, aber keine oder wenig private Eigenschaften und insbesondere über keine auf diesen Eigenschaften arbeitenden Methoden. Daher kann die Konstruktion in eine abstrakte Fabrik verlegt werden. Vorteil ist es für alle generischen Eingabefelder nur noch eine Klasse (und eine Datei) im Verzeichnis zu finden. Nachteil ist die Tatsache, dass diese Klasse im Falle von Erweiterungen verändert werden muss und nicht einfach weitere Klassen im Verzeichnis abgelegt werden. Es überwiegen die Vorteile.

3.3 Iteration C3

Analysis & Design

Funktionsreduktion

Die gewählte Architektur der Kontakteverwaltung muss überarbeitet werden. Das System verwendet Elemente eines sozialen Netzwerkes, Benutzer fügen sich gegenseitig ihrem Adressbuch hinzu. Dies ist nicht gefordert und widerspricht dem geplanten Einsatzzweck.

Nach den Formulierungen des Lastenheftes soll ein Teilnehmer vom Administrator in sein Adressbuch eingetragen werden mit den Parameter:

1. Titel, Name und Vorname
2. E-Mailadresse
3. Telefonnummer
4. Büroadresse
5. Frei definierter Gruppenbezeichnung

Es wird daher in das Datenbankschema eine weitere Tabelle **`contacts`** zur Aufnahme der selbstdefinierten Teilnehmer aufgenommen. Für den generischen Ansatz des Projektes ist die Büroadresse zu spezifisch, es wird daher neben den oben genannten Feldern ein Feld **`comment`** für Bemerkungen eingerichtet.

Implementation

Konsolidierung

Die Änderung der Kontaktestruktur macht etliche Änderungen an verschiedenen Stellen des Codes erforderlich:

1. In der Datenbank wird das Datenbankschema um eine weitere Tabelle

ergänzt.

2. Das *Business-Object-Model* wird um eine Klasse **`Contact`** für die Speicherung eines Kontaktes erweitert.
3. Die Eingabeverifikation ist Teil der Transportobjekte. Die obligatorischen Parameter Vorname, Nachname und E-Mailadresse müssen auf Existenz und Zeichenlänge verifiziert werden.
4. Die Maske zum Hinzufügen von Kontakten wird von einer Suchmaske (für registrierte Benutzer) in einer Eingabemaske verändert. Der Aufwand ist unerwartet gering, weil die Fabrikmethoden zur Erstellung von Formularelementen ausreichend generisch verfügbar sind.
5. Die Datenbankabstraktionsschicht wird um eine Methode zur Speicherung eines Kontaktes erweitert.
6. Das Laden von Kontakten bei Aufruf des Adressbuches muss auf die neuen Datenstrukturen angepasst werden.
7. Es muss eine Funktion zum Löschen von Datensätzen in der Datenbankabstraktionsschicht erstellt und diese Funktion mit der PHP-Seite des Adressbuches verknüpft werden.

Erst nach diesen Vorarbeiten kann die eigentliche fachliche Änderung an der Anwendung vollzogen werden, nämlich die Einladung von selbstangelegten Kontakten und nicht mehr von registrierten Benutzern.

Insgesamt ergibt sich ein Aufwand von zehn Stunden für einen sehr sytemerfahrenen Programmierer. Es zeigt sich, dass späte Änderungen an der Architektur große Aufwände zur Folge haben. Das Beispiel ist darüber hinaus ein klassisches für ein *Feature-Driven-Development*. Die neue Funktion *Einladen von selbstdefinierten Kontakten* wurde an einem Stück quer durch die Architekturschichten implementiert.

Test

Beta-Test

Am Benutzerakzeptanztest waren drei Probenden beteiligt von denen zwei das System zuvor nicht gesehen haben. Die Verbesserungsvorschläge wurden in dem Issue-Tracking-Tool *Mantis* erfasst und in der Anwendung verbessert. Zum Zeitpunkt der Abgabe des Projektes sind noch etwa zehn Probleme der Software gekannt.

4. Transition-Phase

Die Aufwände dieser Phase sind in verschiedenen Projekten sehr unterschiedlich. In diesem Projekt ist diese Phase besonders kurz. An dieser Stelle sollen nur zwei Aspekte genannt werden.

4.1 Implementation

Um das System zu installieren, muss eine einfache Linux-Serverumgebung mit einem installierten PHP-Servermodul und einer SQL-Datenbank gegeben sein. PHP sollte mindestens in der Version 5 und beispielsweise MySQL in der Version 4 installiert sein. Der Authorisationsmechanismus funktioniert nicht mit der CGI-Version von PHP. Nachdem die Adresse des Datenbankservers, die Datenbank und ein Benutzer in den Konfigurationsdateien des Projektes hinterlegt worden sind, fehlt noch die Bereitstellung der Stammdaten.

Dazu wird in der Disziplin *Implementation* eine Routine in die Initialisierungsmodule der Software eingebaut, welche das Vorhandensein der Stammdaten überprüft. Sollten Sie fehlen, so wird automatisch ein Skript zur Bereitstellung der Daten geladen. SQL erlaubt es, dass der eingerichtete Benutzer zwar die Rechte zum Lesen und Schreiben von Daten besitzt, aber nicht zur Veränderung der Struktur. Dann muss das automatische Einrichten der Stammdaten scheitern. Rogatio bemerkt dies und fordert den Benutzer zur manuellen Einrichtung der Daten auf. Erst wenn die Stammdaten bereitstehen, startet das System.

4.2 Environment

Zur Benutzung der Testumgebung und zur Installation auf einem Fremdsystem erstellt die Disziplin *Environment* eine Anleitung. Dies ist die Grundlage für einen *Developer-Guide*, der später geschaffen werden soll. Das Zwischenergebnis als eigenes Kapitel in diese Arbeit eingeflossen.

Teil VI Handbuch

Das Projekt ***rogatio*.net** hat es sich zur Aufgabe gemacht, öffentliche Meinung zu erfassen und zu dokumentieren. Das Projekt bietet kostenlos die Möglichkeit eine Abstimmung zu initiieren und Stimmberechtigte einzuladen.

Falls Sie neu auf dieser Seite sind, sollten Sie einen Blick auf das Kapitel Schnellstart werfen. Alle Details finden sich in der Langversion.

1. Schnellstart

Der folgende Text liefert eine Beschreibung, um auf dem kürzesten Weg zu einer fertigen Abstimmung samt einem aussagekräftigen Stimmergebnis zu kommen. Zusätzliche Funktionen beispielsweise zur Anlage einer Zugangssicherung werden hier ausgespart.

Der Schnelleinstieg ist für Erstbesucher der Seite konzipiert. Er startet mit der Registrierung und legt anschließend eine Abstimmung an, aktiviert diese und sorgt für die Abgabe von Stimmen.

1.1 Registrieren

Den Navigationsbutton *Registrieren* oben rechts auf der Seite auslösen. Dort geben Sie bitte ein:

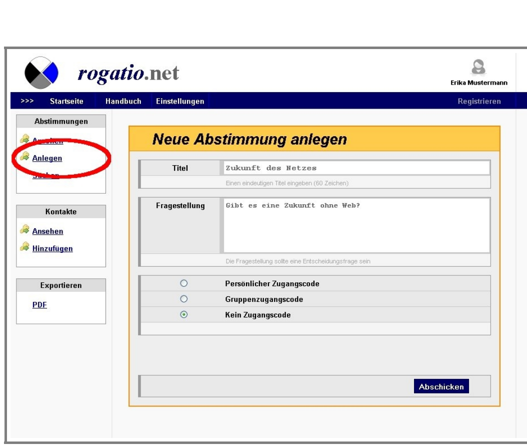
- E-Mailadresse
- Passwort
- Ihren Namen



Ihre E-Mailadresse wird zugleich ihr systemweit eindeutiger Benutzername sein. Zu jeder E-Mmailadresse kann es nur einen Benutzer geben. Aus Gründen des Datenschutzes wird diese Adresse anderen Benutzern nicht angezeigt. Daher ist die Angabe Ihres vollen Namens zur Kennzeichnung Ihrer Abstimmung obligatorisch.

1.2 Anlegen

In der Rubrik *Abstimmungen* wird der Navigationsbutton *Anlegen* permanent eingeblendet. Lösen Sie ihn aus.



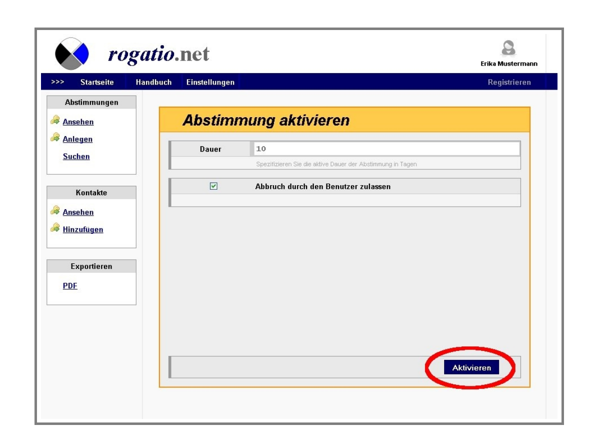
Im folgenden Dialog müssen zwingend definiert werden:

- Titel
- Fragestellung
- Sicherheitsstufe

Der Titel ist auf eine Länge von sechzig Zeichen beschränkt. Er muss definiert werden und er ist pro Benutzer des Systems eindeutig. Die Fragestellung ist der eigentliche Gegenstand der Abstimmung. Sie sollte entsprechend nach ihrem Charakter als Entscheidungsfrage formuliert sein. Als

Sicherheitsstufe wählen Sie bitte *Kein Zugangscode*.

1.3 Aktivieren



Bevor Stimmen für Ihre Abstimmung abgegeben werden können, muss diese aktiviert werden. Lösen Sie dazu bitte den Button *Aktivieren* in der Rubrik *Abstimmungen* aus.

Im folgenden Dialog definieren Sie die Dauer der Abstimmung in Tagen. Es können nur Zeiträume in ganzen Tagen verwendet werden.

Achtung: Die Option *Abbruch durch den Benutzer zulassen* erlaubt jedem Stimmberechtigten Ihre Abstimmung abzubrechen.

1.4 Abstimmen



Nach der Aktivierung können die Stimmberechtigten abstimmen. Lösen Sie dazu den Button *Abstimmen* in der Rubrik *Abstimmungen* aus. In der gewählten Sicherheitsstufe sind alle Besucher der Webseite stimmberechtigt, die geschaffene Abstimmung hat den Charakter eines TED.

Im nachfolgenden Kapitel mit dem Titel Langversion erfahren Sie, wie eine Abstimmung durch Gruppenzugangscodes oder individuelle Zugangscodes geschützt werden kann.

1.5 Ansehen



Sie sehen nach Ihrer Stimmabgabe die Veränderung der Sicht auf die Abstimmung. Das Diagramm zeigt die prozentuale Verteilung der Stimmen und die Legende die absoluten Zahlen. In der Statuszeile unterhalb des Diagrammes ist das planmäßige Ende der Abstimmung aufgeführt.

Sie können jede Abstimmung

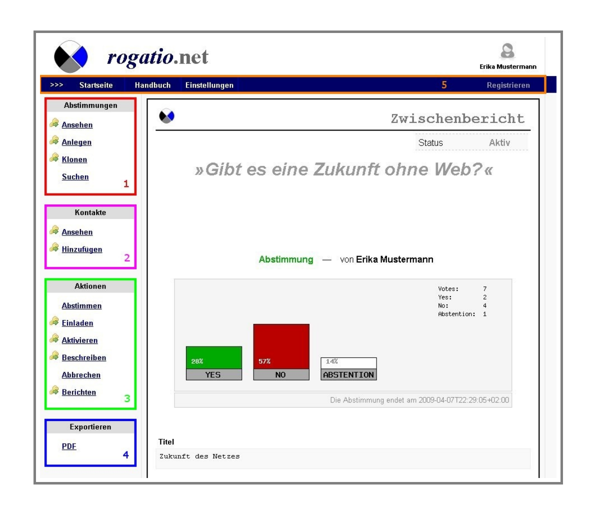
jederzeit wieder aufrufen, indem Sie entweder die Suchmaske verwenden oder über den Button *Ansehen* eine komplette Liste Ihrer Abstimmungen anfordern.

2. Langversion

Dieses Kapitel ist der Kern des Benutzerhandbuches des Projektes ***rogatio*.net**. Es gliedert sich in drei wesentliche Teile. Zunächst wird ein ausführlicher Überblick über die Navigation und alle verfügbaren Optionen zur Steuerung der Anwendung gegeben. Der zweite Abschnitt erläutert die darstellende Oberfläche, also insbesondere die Bedeutung der verschiedenen Informationen einer Abstimmungsseite. Der letzte Abschnitt ist ein Blick hinter die Kulissen. Er erläutert einige Basiskonzepte, die bei der Entstehung der Anwendung Pate gestanden haben.

2.1 Navigation

Die Navigation gliedert sich in die Systemnavigation einerseits und thematisch vier verschiedene Rubriken andererseits. Die Systemnavigation umfasst die Punkte der horizontalen Menüleiste. Sie haben globalen Charakter und beziehen sich auf systemweite Themen. Die Leiste ist in der Übersicht orange gekennzeichnet.



Rubriken

Für die Steuerung der rot gekennzeichneten Abstimmungen und der Kontakte – violett – gibt es jeweils eigene Rubriken, die dem Benutzer permanent zur Verfügung stehen. Im Gegensatz dazu steht die Rubrik Aktionen, in der Übersicht grün gekennzeichnet. Sie wird kontextsensitiv eingeblendet, aktuell insbesondere bei der Ansicht einer Abstimmung. Die angebotenen Optionen beziehen sich immer auf die konkret dargestellten Daten. Auch die Optionsfläche Nummer vier – Export – ist kontextsensitiv. Der angebotene Datenexport in das Portable-Document-Format (PDF) generiert eine Seite im Format DIN-A4 des naheliegensten Berichtes – einer Abstimmung, einer Übersicht oder einer Handbuchseite. [Der Export steht in der Alpha-Version der Software noch nicht zur Verfügung, Anm. d. Red.].

Schlüsselsymbol



Für einige Aktionen benötigt das System Kenntnis über den Administrator, beispielsweise zur Anlage einer Abstimmung oder dem Hinzufügen eines Kontaktes. Dazu wird ein Mechanismus der impliziten Authorisierung verwendet: Beim Aufruf der Seite erscheint ein browsergesteuertes Authorisationsfenster. Bitte geben Sie hier zunächst Ihren Benutzernamen und Ihr Passwort ein. Sie werden dann automatisch auf die angewählte Seite weitergeleitet. Eine gesondere Funktion zum Anmelden (Login) ist damit nicht mehr notwendig. Die Seiten, die eine Authorisierung erfordern sind mit einem Schlüsselsymbol gekennzeichnet, siehe Screenshot rechts.

Systemnavigation



Die Systemnavigation bietet Optionen zur globalen Steuerung des Systems. Alle Optionen sind in der horizontalen Menüleiste unterhalb der Kopfzeile untergebracht. Alle Optionen sind Verknüpfungen – HTML-Anchors – gerichtet auf Seiten des Projektes. Die Verweise auf die Startseite und das Handbuch sind selbsterklärend.

Modalität

Für die Verwendung des Handbuches kann optional ein nicht-modales zusätzliches Fenster geöffnet werden. Prinzipiell öffnen sich alle Verweise per Vorgabe im gleichen Fenster. Trick: Wenn Sie das Handbuch als parallele Hilfestellung in der Navigation verwenden wollen, so können Sie mit einem Klick bei gedrückter Steuerungstaste (STRG) das Handbuch in einem neuen Fenster (oder Karteireiter) öffnen. Dies ist eine Funktion Ihres Browsers, aber alle wichtigen Browser unterstützen diese Möglichkeit.

Einstellungen



Hinter dieser Option verbergen sich Einstellungen, die sie applikationsweit vornehmen können. Aktuell ist dies nur die Sprachvorgabe. Unterstützt werden Deutsch (DE-DE) und Englisch (EN-US). Ihre Auswahl wird in der *Session* gespeichert, sie ist nach dem Schließen des Browser daher verloren. Erst zukünftige Versionen von ***rogatio*.net** werden eine permanente Speicherung ermöglichen. Nach dem Speichern der Sprachvorgabe zeigt das System wieder die Startseite.

Registrierung



Zur Registrierung geben Sie bitte Ihre E-Mailadresse, ein Passwort und Ihren Namen an. Ihre E-Mailadresse wird zugleich ihr systemweit eindeutiger Benutzername sein. Zu jeder E-Mmailadresse kann es nur einen Benutzer geben. Aus Gründen des Datenschutzes wird diese Adresse anderen Benutzern nicht angezeigt. Daher ist die Angabe Ihres vollen Namens zur Kennzeichnung Ihrer Abstimmung obligatorisch.

Abstimmungen

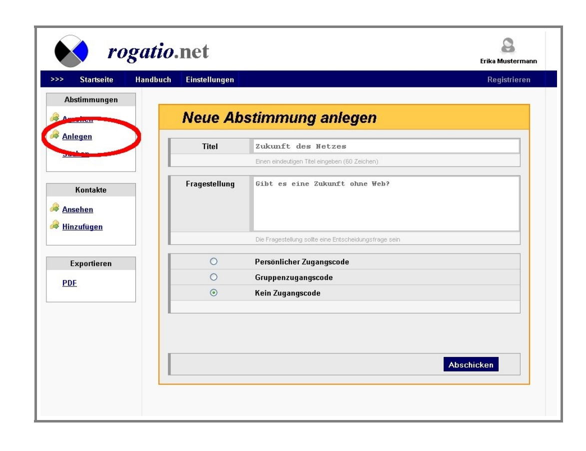


Die Rubrik *Abstimmungen* wird permanent eingeblendet. Sie zeigt allgemeine Optionen für Abstimmungen, die keinen Bezug zu einer konkreten Abstimmungen haben.

Ansehen

Hinter dem Verweis *Ansehen* verbirgt sich die Liste aller eigenen Abstimmungen. Sie werden chronologisch sortiert auf unter Umständen mehreren Seiten angezeigt. Sie können zwischen den Seiten hin- und herblättern.

Anlegen



Eine Abstimmungen besteht aus einer Reihe von Parametern, die zum überwiegenden Teil vom Administrator einer Abstimmung vorgegeben werden. Das Minimum an definierten Parametern einer Abstimmung ist die Kombination aus einem Titel, der Fragestellung und der Definition der Zugangssicherungsschärfe. Dieses Minimum muss bei der Erstellung zwingend spezifiziert werden, weitere Parameter werden im weiteren Verlauf der Konfiguration einer Abstimmung hinzugefügt.

- **Titel:** Der Titel darf nicht leer bleiben und kann aus höchstens 60 Zeichen bestehen. Der Titel einer Abstimmung ist unter der Summe aller Abstimmungen eines Administrators eindeutig. Sie dürfen also einen Titel nicht zweimal vergeben – das schließt natürlich nicht aus, dass ein anderer Benutzer des Systems Ihren Titel ebenfalls verwendet. Hinweis: Dieser Schutz ist notwendig, um doppelte Einträge in der Datenbank zu verhindern.
- **Fragestellung:** Für eine Abstimmung ist natürlich eine Frage erforderlich, sie stellt ja gerade den Abstimmungsgegenstand dar. Sie darf nicht mehr als 160 Zeichen umfassen. Ihre Abstimmung kann von den Stimmberechtigten mit den Antworten Ja, Nein oder einer Enthaltung belegt werden. Sie sollten bei der Formulierung der Frage daher darauf achten, dass sie mit diesen Antworten sinnvoll belegt werden kann.
- **Zugangscode:** Eine Abstimmung kann vor unberechtigtem Zugriff geschützt werden. Es muss sichergestellt werden, dass nur stimmberechtigte Besucher der Seite ihre Stimme auch abgeben können. Dies geschieht in ***rogatio*.net** durch die Verwendung von Zugangscodes. Mit dem Schutz des individuellen Zugangscodes wird für jeden Stimmberechtigten ein eigener Code generiert, der ihm in seiner Abstimmungsaufforderung per E-Mail mitgeteilt wird. Um auch eine ganze Gruppe von Stimmberechtigten über eine E-Mailverteiler einladen zu können, gibt es die Option des Gruppenzugangscodes. Dieser ist für alle Emüpfänger der Abstimmungsaufforderung identisch. Eine Weitergabe des Codes kann damit natürlich vom System nicht ausgeschlossen werden. Für Abstimmungen mit dem Charakter eines TED – im Webumfeld nach dem Netzjargon auch Web-Poll genannt – kommt auch der Verzicht auf einen Zugangscode in Frage. Stimmberechtigt sind damit prinzipiell alle Besucher der Seite.

Suchen



Das System bietet die Möglichkeit nach Abstimmungen gezielt zu suchen. Im Suchfeld kann dazu ein Teil des Titels genannt werden. Das System listet anschließend alle Abstimmungen auf, die diesen Text im Titel führen. Die Suche beachtet nicht die Groß- und Kleinschreibung (nicht case-sensitiv). Gesucht wird tatsächlich in allen Abstimmungen von allen registrierten Administratoren. Dies ist eine Möglichkeit auch fremde Abstimmungen einzusehen und gegebenenfalls dort eine Stimme abzugeben.

Kontakte

Die Rubrik Kontakte wird permanent eingeblendet. Sie haben hier die Möglichkeit eine Übersicht der bisher eingerichteten Kontakte aufzurufen, oder einen neuen Kontakt hinzuzufügen. Jeder Benutzer hat sein eigenes Verzeichnis von Kontakten.

Hinzufügen

Um einen neuen Kontakt anzulegen, müssen Sie zwingend die Felder Vorname, Nachname und E-Mailadresse ausfüllen. Das Ausfüllung der restlichen Felder ist optional.

- **Name:** Die Angaben zum Namen umfassen zwingend einen Vor- und Nachnamen. Die Angaben können nicht leer bleiben und können jeweils aus höchstens 50 Zeichen bestehen.
- **E-Mailadresse:** Damit der Kontakt über Ihre Einladung zu einer Abstimmung informiert werden kann, müssen Sie seine E-Mailadresse kennen.
- **Titel:** Das Feld Titel dient der Vergabe von Ehren- oder akademischen Titeln. Dies ist optional.
- **Telefonnummer:** Unter diesem Stichwort haben Sie die Möglichkeit eine Telefonnummer des Kontaktes abzulegen. Das Feld ist ein Textfeld und wird nicht validiert. Der Text wird nicht in ein Standardtelefonnummernformat umgewandelt.
- **Gruppenbezeichner:** Eine Gruppenbezeichnung kann von Ihnen frei vergeben werden. Sie können den Kontakt damit einer Rubrik zuordnen. Spätere Version von ***rogatio*.net** werden eine Filterung nach diesem Gruppenbezeichner ermöglichen.

- **Bemerkung:** Im Feld Bemerkung können Sie nach Belieben weitere Informationen ablegen. Es ist aus der Anforderung nach der Ablage einer Büronummer entstanden und kann beispielsweise diesem Zweck dienen.

Aktionen



Die Rubrik zeigt die **Aktionen** kontextspezifisch an. Insbesondere für Steuerung und weitere Parameterisierung einer Abstimmung stehen zahlreiche Verweise zur Verfügung. Alle Aktionen beziehen sich immer auf das konkret dargestellte Element. Über die Verweise kann die Abstimmung entweder weiter parameterisiert oder in Ihrem Zustand verändert werden.

Abstimmen



Hier kann der zentrale Vorgang der Anwendung vorgenommen werden, die Abgabe der Stimme. Falls die Abstimmung mit einem Zugangscode geschützt ist, so muss dieser neben dem Ankreuzen selbst auch in der Abstimmungsmaske belegt werden. Ohne gültigen Code wird die Stimme nicht in die Datenbasis übernommen. Individuelle Codes erlauben nur einmal die Abgabe der Stimme mit diesem Code.

Einladen



Dieser Verweis erlaubt die Einladung von Stimmberechtigten. Ist die Abstimmung mit einem individuellen Zugangscode geschützt, so ist die Einladung obligatorisch. Dies ist notwendig, weil nur dann für jeden Stimmberechtigten ein Code generiert werden kann.

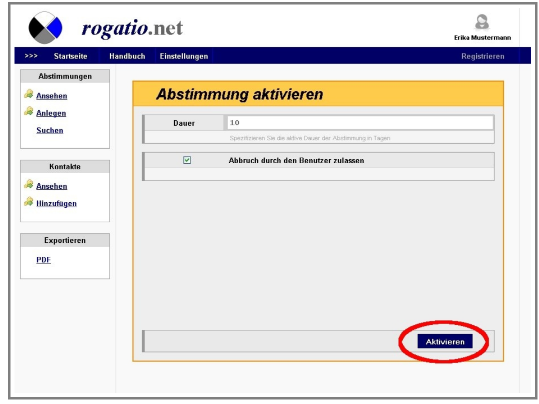
Bei Verwendung eines Gruppenzugangscodes ist die Einladung von Stimmberechtigten ebenfalls empfohlen. Diese bekommen eine automatisch vom System verschickte

Benachrichtigung per E-Mail mitsamt dem Zugangscodes.

Ohne Zugangscode ist die Einladung optional, sie kann zur Information von potentiell Interessierten genutzt werden, denn auch in diesem Fall werden Benachrichtigungen verschickt.

Aktivieren

Um für eine Abstimmung tatsächlich abstimmen zu können, muss diese zunächst vom Administrator aktiviert werden. Dadurch können Sie Ihre Abstimmung in aller Ruhe konfigurieren und erst danach freigeben. Zur Freigabe wird die Dauer der Abstimmung und die Erlaubnis zum Abbruch vergeben.



- **Dauer:** Eine Abstimmung hat in ***rogatio*.net** immer eine Dauer von einer Summe von ganze Tagen. Diese Tageszahl muss größer Null und kleiner als zehn Jahre sein.
- **Benutzerabbruch:** Sie können durch die Aktivierung der Option *Abbruch durch den Benutzer zulassen* erlauben, dass jeder Stimmberechtigte die Abstimmung unter der Vergabe eines Kommentars abbrechen kann. Die Abstimmung ist danach in einem eigenen Zustand *Abgebrochen* und kann nicht mehr verwendet werden. Achtung: Ohne die Verwendung eines Zugangscodes ist jeder Besucher Ihrer Abstimmung auch stimmberechtigt.

Beschreiben

Zu jeder Abstimmung können Sie optional eine Beschreibung eingeben. Während die Fragestellung mit maximal 160 Zeichen relativ kurz ist, können Sie hier ausführlicher das Anliegen der Abstimmung beschreiben. Sie können die Beschreibung formatieren. Als Formatierungssprache kommt BB-Code zum Einsatz. Weitere Details zu den Formatierungsmöglichkeiten finden Sie im Abschnitt Formatierung des Kapitels Konzepte. Die Beschreibung einer Abstimmung darf eine Länge von bis zu 800 Zeichen umfassen. Ein einzelnes Wort der Beschreibung sollte aus Gründen der Darstellung jedoch 50 Zeichen nicht überschreiten.



Abbrechen

Wenn der Administrator einer Abstimmung es erlaubt, so können Stimmberechtige statt der Abgabe Ihrer Stimme, die Abstimmung auch vorzeitig abbrechen. Damit wird die Möglichkeit eröffnet, einen Beratungsbedarf anzumelden, der über die Möglichkeiten von ***rogatio*.net** oder andere elektronischen Hilfsmitteln hinausgeht.



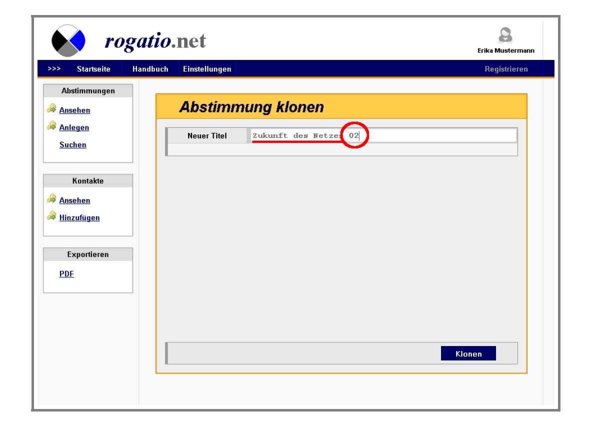
Berichten

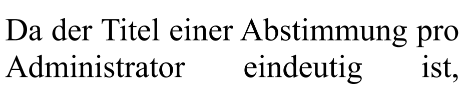
Die Aktion *Berichten* ermöglicht es dem Administrator einer Abstimmung, über die internen Kommunikationssysteme von ***rogatio*.net** eine Nachricht mit dem Ergebnis oder einem Zwischenergebnis der konkreten Abstimmung zu versenden. Der Bericht kann zudem mit einem Kommentar versehen werden.



Klonen

Eine Abstimmung, die sich in den Status **Beendet** oder **Abgebrochen** befindet, kann nicht mehr reaktiviert werden. Um die Daten aber dennoch weiterhin verwenden zu können oder eine abgebrochene Abstimmung nach erfolgter Beratung neu aufnehmen zu können, existiert die Option zum klonen.





Da der Titel einer Abstimmung pro Administrator eindeutig ist, müssen Sie einen neuen Titel definieren. Dies können sie tun, indem Sie beispielsweise eine laufende Nummer an den Titel anhängen.

Export

Die Rubrik **Export** ist wie die Rubrik Aktionen immer im Zusammenhang mit den angezeigten Elementen zu sehen. Durch den Verweis wird vom System ein PDF dynamisch mit den dargestellten Inhalten erzeugt.

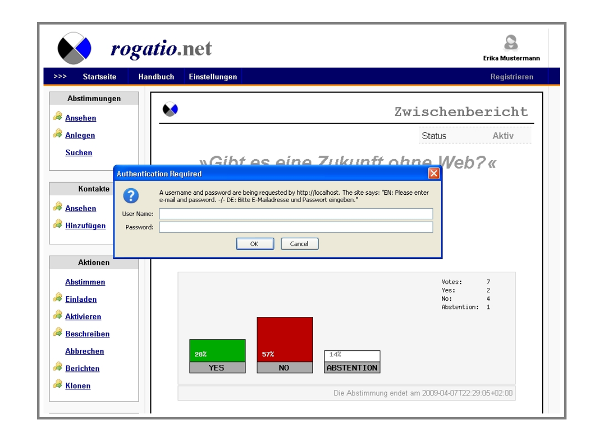
[Der PDF-Export steht in der Alpha-Version von ***rogatio*.net** noch nicht zur

Verfügung, Anm. d. Red.]

2.2 Konzepte

In diesem Kapitel sollen einige der Schlüsselkonzepte der Anwendung vorgestellt werden. Es wird kein Anspruch auf Vollständigkeit erhoben.

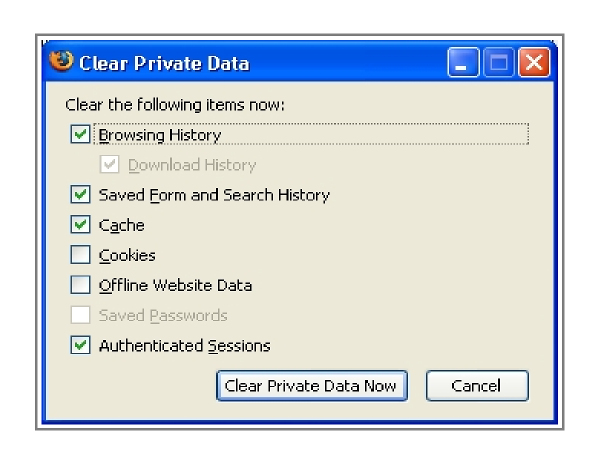
Authorisierung



Das Projekt ***rogatio*.net** verwendet eine Form der Authorisierung, welche für Webanwendungen ungewöhnlich ist, aber zum Verzeichnisschutz auf Webservern seit langem zum Einsatz kommt: HTTP-Authorisierung.

Diese gleicht einen Nachteil bekannter Anmeldemechanismen aus, bei denen Passwörter im Klartext übertragen werden und leicht mitgelesen werden können. Die HTTP-Authorisierung wird in der *Digest*-Variante eingesetzt und gilt als sicher.

Mit diesem Verfahren ändert sich der Umgang mit der Anwendung. Es ist nun möglich, zunächst einmal einen Zugang ohne Login zu realisieren und erst beim Aufruf einer Seite, die eine Authorisierung benötigt, diese auch anzufordern. Dies geschieht implizit durch den Browser, nicht über einen separaten Bildschirm. Nahezu alle Browser unterstützten das Verfahren. Eine Ausnahme ist lediglich der textbasierte Browser Lynx bekannt.



Es ist in der HTTP-Spezifikation kein Logout vorgesehen. Dadurch ist es möglich eine einmal erfolgte Anmeldung permanent aufrecht zu erhalten. Soll das System an einem Browser von unterschiedlichen Teilnehmern benutzt werden, so reicht ein Schließen des Browsers zum Logout. Mozilla Firefox bietet unter *Tools*, *Clear Privacy Data*, *Authenticated Sessions* sogar die Möglichkeit zu einem manuellen Logout ohne Schließen des Browsers an.

Formatierung

In einigen Eingabefeldern ist eine Formatierung der Daten möglich. Verwendet wird dazu eine Auszeichnungssprache, welche sich in frühen Webforen (englisch

Bulletin Board) entwickelt hat und daher unter dem Namen BB-Code bekannt ist. Das Verfahren hat eine große Ähnlichkeit zur sehr frühen HTML-Textauszeichnung, aber es ist nicht standardisiert und wird von verschiedenen Forensystemen, Wikis und Content-Management-Systemen unterschiedlich umgesetzt. Das Projekt ***rogatio*.net** verwendet aktuell eine Teilmenge von BB-Code, welche allgemein verbreitet sein sollte. Jede Form von direkter Formatierung durch HTML wird vom System aus Sicherheitsgründen ausgefiltert.

Schriftformatierung

`[b]Text[/b]` schreibt fett (**`Text`**).

`[u]Text[/u]` schreibt unterstrichen (`Text`).

`[i]Text[/i]` schreibt kursiv (*`Text`*).

`[c]Text[/c]` zentriert den Text.

Listen

Auch Listen werden analog zu HTML umgesetzt. Aus

```
[list]
[*]Erster Eintrag[/*]
[*]Zweiter Eintrag[/*]
[/list]
```

wird die Liste

- Erster Eintrag
- Zweiter Eintrag

Einsatzbereich

Aktuell kann die Beschreibung von Abstimmungen durch BB-Code formatiert werden.

Status

Eine Abstimmung durchläuft einen definierten Lebenszyklus. Der Zyklus ist nicht linear, aber auch nicht zyklisch. Der Status einer Abstimmung bestimmt welche Aktionen gestartet werden können. Folgende Status sind möglich:

- **Unvollständig:** Die Abstimmung wurde erstellt, aber bis zur Aktivierung fehlen noch weitere Parameter. In diesem Zustand befindet sich eine Abstimmung mit individuellen Zugangscodes, wenn noch keine Teilnehmer eingeladen worden sind.
- **Ausführbar:** Dies ist eine Abstimmung, die bereit ist, aktiviert zu werden, aber noch nicht aktiviert worden ist.
- **Aktiv:** Eine Abstimmung, für die von den Stimmberechtigten abgestimmt werden kann. Sie wurde vom Administrator aktiviert, ihre Dauer ist noch nicht abgelaufen und sie wurde nicht abgebrochen.
- **Abgebrochen:** Dieser Status signalisiert, dass die Abstimmung nicht ihr reguläres zeitliches Ende erreichen konnte. Sie wurde vom Administrator

abgebrochen, oder von einem Stimmberechtigten.
- **Beendet:** Mit diesem Status hat die Abstimmung ihr definiertes zeitliches Ende regulär erreicht.

Zugangscode

Mit Zugangscodes wird der Zugang von Stimmberechtigten zu einer Abstimmung kontrolliert. Während gemäß der Architektur des Projektes ***rogatio*.net** jede Abstimmung prinzipiell von jedem Besucher einsehbar ist, soll das Abstimmungsverhalten durch den Administrator einer Abstimmung kontrollierbar sein. Dazu werden bei der Erstellung einer Abstimmung folgende drei Optionen angeboten:

- **Individueller Zugangscode:** Für jeden Stimmberechtigten wird ein eigener Code generiert. Dazu müssen vor der Aktivierung der Abstimmung Stimmberechtigte aus dem Adressbuch oder per E-Mail eingeladen worden sein.
- **Gruppenzugangscode:** Der Gruppenzugangscode ist ein Code, der global für alle Stimmberechtigten identisch ist. Damit können Interessierte beispielsweise über einen E-Mailverteiler eingeladen werden. Es spricht auch nichts gegen die Weitergabe des Codes auf anderen Wegen, wie dem Postweg. Das System kann allerdings nicht sicherstellen, dass der Code auch an Unberechtigte weitergegeben wurde. Zukünftige Versionen von ***rogatio*.net** werden erlauben, dass die Anzahl von Stimmberechtigten definiert wird, um dieses Problem abzumildern.
- **Kein Zugangscode:** Wird kein Zugangscode verwendet, so wird jeder Besucher der Abstimmung zum Stimmberechtigten. Auf diese Weise lassen sich Umfragen im Stile eines TED – oder Web-Poll oder Videotextumfrage – realisieren. Durch Erschweren von Mehrfacheingaben (siehe Abschnitt Gleichheit[141]) kann dies das zweckmäßigeste Verfahren sein.

Gleichheit

Mit der Verwendung von Zugangscodes kann sichergestellt werden, dass nur stimmberechtigte Nutzer ihre Stimme abgeben können. Eine Mehrfachvergabe Ihrer Stimme – absichtlich oder aus Versehen – ist aber nur bei der Verwendung von individuellen Zugangscodes ausgeschlossen. Um auch die anderen Verfahren zu sichern stellt ***rogatio*.net** zwei Möglichkeiten bereit.

1. **Sicherung der IP-Adresse:** Mit diesem Verfahren soll sichergestellt werden, dass einfaches zweimaliges Abgeben einer Stimme, möglicherweise durch wiederholtes Senden des Requestes durch Aktualisierung der Seite (F5 drücken) verhindert wird. Besucher mit einer unveränderlichen IP-Adresse werden damit zunächst an Manipulationen gehindert. Achtung: Viele Surfer verwenden eine von ihrem Provider dynamisch vergebene Adresse. Diese wird bei jeder Neueinwahl vergeben. Damit wird der Schutz unwirksam. Mit der Aktivierung der IP-Sicherung wird lediglich eine relative Hürde für den Nutzer gesetzt. Auch beim

141 Siehe Kapitel III.4.

Wechsel des Computers beispielsweise in einer öffentlichen Einrichtung wird der Schutz umgangen. Trotzdem wird die allereinfachste Manipultionsmöglichkeit im per HTTP-Resend (Browser-Aktualisierung) verhindert. Da ein ehrlicher Nutzer durch das Tracking der IP-Adresse keinen Schaden nimmt, spricht grundsätzlich nichts gegen die Aktivierung der Option. Der Administrator muss diese Überlegungen gegen seine datenschutzrechtlichen Prinzipien halten.

2. **Vermerken in Session-Variablen:**Bei Aktivierung dieser Option speichert das System in einer Session-Variablen, dass der Nutzer seine Stimme bereits abgegeben hat. Auch damit wird eine Manipulation durch einfaches erneutes Laden der Seite unterbunden. Die Beurteilung der Schutzwirkung dieser Maßnahme hängt von den Kenntnissen des Benutzers ab. Session-Variablen werden vom ***rogatio*.net** zugrundeliegenden PHP-System versucht in Cookies zu speichern. Der Benutzer kann diese ohne Kontrolle des Systems löschen und damit auch die Werte der Session entfernen. Sollte der Benutzer die Verwendung von Cookies sogar deaktiviert haben, so kommt ein eine Session-Identifikationsnummer zum Einsatz, welche bei jedem HTTP-Request zwischen Server und Client ausgetauscht wird. Das Prinzip ist nicht zur Kontrolle des Benutzers entworfen und so gibt es für den technisch informierten Benutzer viele Möglichkeiten den Vermerk seiner Abstimmung unwirksam zu machen. Daher ist auch dieser Schutz nur eine unterstützende Maßnahme.

[Die beiden Funktionen *Sicherung der IP-Adresse und* und *Vermerken in Session* sind in der Alpha-Version von ***rogatio*.net** nicht implementiert, Anm. d. Red.]

Teil VII Installation

Der folgende Abschnitt soll ein kurze Anleitung zur Installation der Anwendung auf einem Webserver und zur Benutzung der lokalen Testumgebung liefern.

Achtung: Die Software hat in ihrem Charakter als akademisches Projekt keine ausreichende Sicherheitsvorgaben, um sie auf einem produktiven Server zu betreiben. Zu diesem Zweck müssten Administrationsverzeichnisse zunächst entsprechend vor fremden Zugang geschützt werden. Dieser Schutz ist nicht aktiviert, um einen guten Einblick in die Datenarchitektur gewährleisten zu können.

1. Webserverinstallation

Zum Betrieb der Software ist eine sehr gewöhnliche Linux-Serverumgebung notwendig, nämlich PHP 5 und MySQL 4. Andere SQL-Dialekte sollten ebenfalls funktionieren, das System verwendet keine spezifischen Befehle. Die einzige spezielle Anforderung ist, dass PHP zwingend als Servermodul installiert werden muss. Die alternative Installation als CGI-Modul funktioniert nicht mit der HTTP-Authentifizierung. Die Serverinstallation aller Provider, die normale Webpakete anbieten beinhaltet PHP als CGI-Modul, weil bei der Installation als Apache-Servermodul PHP die gleichen Rechte auf dem Gastrechner inne haben würde, die der Webserver auch hat. Daher kommt für die Installation nur ein eigener Server in Frage, kein Webpaket. Getestet wurde allerdings ein virtueller Server, wie er beispielsweise von der Firma Server4You angeboten wird. Die Installation auf einem Windows Server wie dem IIS wurde nicht getestet. Anleitung zur Installation:

1. Software auf den Server kopieren
2. In der Datei `access/DatabaseAccess.php` die Zugangsdaten zur Datenbank eintragen.
3. Die Datei `index.php` starten.

Das System schreibt Fehlermeldungen auf den Bildschirm, wenn es den Datenbankserver nicht finden oder die Datenbank nicht öffnen kann.

Falls die Datenbank leer ist, versucht das System die notwendigen Stammdaten in die Datenbank einzuspielen. Sollte dies nicht funktionieren, weil die Rechte des hinterlegten Benutzers beispielsweise nur zum Lesen und Schreiben von Daten, aber nicht zur Manipulation von Strukturen ausreichen, so fordert das System zur manuellen Einrichtung der Stammdaten auf. Dazu muss lediglich die Daten `schema/structure.sql` in die erstellte Datenbank importiert werden. Nun kann das System arbeiten. Es werden Laufinformationen in der Datei `logging/trace.log` gespeichert.

2. Lokale Testumgebung

Mit der Software wird eine komplette Test- und Entwicklungsumgebung geliefert, die bereits zur Erstellung genutzt worden ist. Sie besteht aus portabler Software, muss also zur Ausführung lediglich auf einen Rechner mit einem Windows-Betriebssystem kopiert werden. Die Ausführung direkt von einer CD ist nicht möglich, weil zum Start des Webservers Daten geschrieben werden müssen. Getestet wurde die Umgebung unter Windows XP. Die Benutzung der Testumgebung ist empfohlen. Die einzige Einschränkung dieser Umgebung ist, dass der Versand von E-Mails nicht möglich ist. Eine (befüllte) Datenbank, phpMyAdmin und der Bugtracker Mantis stehen ebenfalls zur Verfügung. Die beigefügte Firefox-Browser enthält Lesezeichen auf die relevaten URLs. Die Release-Version wird bereits beim Start des Browsers geladen. Um die Testumgebung zu starten:

- 01_Project/10_Environment/server.vbs
- 01_Project/10_Environment/firefox.vbs

Folgende Pfade führen zu den drei installierten Webanwendungen:

- Mantis: http://localhost/mantis
- phpMyAdmin: http://localhost/phpmyadmin
- Rogatio: http://localhost/rogatio

Für phpMyAdmin ist keine weitere Authorisation notwendig. Für Mantis lautet der Administratorzugang:

- Name: `administrator`
- Passwort: `open`

Teil VIII Anhänge

Die Arbeit hat vier Anhänge, welche die Rahmenbedingungen des Projektes definieren. Es sind eine Aufführung des Pflichtenheftes, des Lastenheftes, der erforderlichen Change-Requests und die Testfälle. Schließlich folgt das Literaturverzeichnis.

1. Lastenheft

Dieser Anhang hat das Lastenheft zum Inhalt. Das Lastenheft ist eine grobe Beschreibung der gewünschten Systemfunktionalität und hat daher Anforderungscharakter. Es beschreibt funktionale Anforderungen, einige Geschäftsprozesse – Business Use-Cases – und Andeutungen der Systemarchitektur.

1.1 Wortlaut

Konzipieren und Entwickeln Sie ein online-e-Voting-System (Abstimmungssystem) zur Nutzung innerhalb eines Intranets (wie zB der Universität). Das Ziel wäre die Unterstützung der Arbeit eines Gremiums, das im Prinzip bei Face-to-Face-Sitzungen Entscheidungen trifft, aber in Ausnahmefällen dies auch per Umlaufbeschluss (via Papier, e-Mail) tut. Es wäre eine Webapplikation (PHP?) zu entwickeln, die eine Schnittstelle A zum Aufruf zur Abstimmung und eine Schnittstelle V zum Abgeben der Stimmen umfasst. Die Applikationen würde das Abstimmen per e-Mail oder Papier ablösen, um die entsprechenden Vorgänge zu erleichtern.

Organisatorisches Umfeld

Die Abstimmenden (Nutzer von A) bestehen aus den Mitgliedern des Gremiums – genau die dürfen abstimmen. Die sind also namentlich bekannt, ihre e-Mail-Adressen sind bekannt. Es gibt einen Vorsitzenden des Gremiums - der darf zu einer Abstimmung aufrufen (Nutzer von V). Alle Abstimmungen sind zeitlich fest begrenzt - und es darf nur innerhalb der gültigen Frist abgestimmt werden.

Schnittstelle A

Der Vorsitzende kann mit Schnittstelle A

- eine neue Abstimmung initiieren, dabei die Abstimmenden auf die neue Abstimmung per e-Mail aufmerksam machen, und ihnen auch gleich den Link auf die Abstimmungsseite mitschicken
- die Parameter der Abstimmung (s.u.) eingeben
- die Abstimmung irgendwann "scharf" schalten - dann darf er die Daten der Abstimmung (Ausnahme: der Kommentar) nicht mehr verändern
- die Ergebnisse einer Abstimmung auslesen und als Report vom Abstimmungssystem entgegennehmen
- das Ergebnis an die Abstimmenden verteilen
- die Mitgliederliste des Gremiums bearbeiten (nur er kann das, nicht die Leute selber)

Eine Abstimmung hat als Parameter

- KB: eine griffige Bezeichnung (Text, eine Zeile)
- LB: eine Langbeschreibung der Thematik (Text, mehrzeilig, einfaches HTML)
- K: ein Kommentar - der darf während einer Abstimmung als einziger noch modifiziert werden
- M: wie bei einer e-Mail die Möglichkeit, Material zum Beschluss als Attachments anzufügen, dies können beliebige Binärdokumente sein (PDF, Word, Excel) oder auch e-Mails
- AA: eine Textaussage (Abstimmungsaussage), über die abgestimmt wird, also der deutsche Satz, dem die Teilnehmer zustimmen können oder den sie ablehnen können, sowas wie "Herrn X sollte ein weiterer Prüfungsversuch zugestanden werden."
- ZG: Zeitgrenzen für die Freischaltung der Abstimmung, diese sollte mindestens die Arbeitszeit zweier aufeinanderfolgender Werktage umfassen (also sowas wie DO 8:00 Uhr bis FR 17:00 Uhr), wobei Samstag hier nicht mitzählen sollte (obwohl es juristisch Werktag ist), dürfte aber auch länger sein

KB, LB, K, M, AA und ZG sollen vom Nutzer der Schnittstelle A direkt eingebbar sein - ZG sollte überprüft werden (Meldung, wenn falsch) - bis auf K darf nichts mehr geändert werden, wenn die Zeit der Abstimmung läuft.

Der Vorsitzende kann Abstimmungen editieren solange sie nicht scharf sind.

Wird sie scharf geschaltet, geht eine e-Mail mit dem Link auf die Schnittstelle A für diese Abstimmung an alle Mitglieder (auch an den Vorsitzenden selbst) in der auf die anstehende Abstimmung unter Angabe des Zeitfensters hingewiesen wird. Im Subject die KB, im Text Standardtext, ZG und der Link. Als reply-to der Vorsitzende.

Eine scharf geschaltete Abstimmung kann der Vorsitzende explizit abbrechen – wenn sich gezeigt hat, dass die Abstimmung keinen Sinn mehr macht. Danach kann er sie dann auch wieder bearbeiten (es wird dann automatisch eine neue) und wieder scharf schalten. Alle "alten" Abstimmungen bleiben erhalten - er sollte eine neue Abstimmung aus einer alten einfach entwickeln können (Copy-Paste).

Wenn die Zeit abgelaufen ist, erhalten alle Mitglieder automatisch eine e-Mail mit dem Rohergebnis der Abstimmung - natürlich auch der Vorsitzende. Das Rohergebnis besteht aus der Angabe

- Wer hat abgestimmt (Anzahl und Namen)
- Wieviele Ja-Stimmen, Enthaltungen, Nein-Stimmen
- Letzteres nur, wenn keine Sitzung gefordert wurde, sonst die Aussage, wer eine Sitzung gefordert hat.

Wenn die Abstimmung gelaufen ist, dann steht auch in der Schnittstelle A das Ergebnis, inhaltlich wie in der o.a. Mail an die Mitglieder und der Vorsitzende kann per Button eine e-Mail an die Mitglieder schicken, die zum Rohergebnis zusätzlich die Inhalte eines Textfelds enthält, in das der Vorsitzende einen Text

schreiben kann. Also zB: "Damit haben wir den Antrag angenommen." oder "Wir waren leider nicht beschlussfähig - es haben zu wenige Mitglieder abgestimmt - ich werde es nächste Woche noch einmal probieren." oder "Die geforderte Sitzung zum Thema findet am .... statt."

Schnittstelle V

Die Abstimmenden sehen auf dieser Seite in entsprechender Gestaltung

- KB, LB, K als Texte
- M als Links zum Downloaden des Materials
- Darunter eine Radio-Button-Einheit mit darüber dem Satz AA, dann vier Alternativen markiert mit "Ich stimme zu", "Ich stimme nicht zu", "Ich enthalte mich", "Ich widerspreche der Abstimmung per Umlaufbeschluss und wünsche eine Sitzung" - ohne Voreinstellung!
- Darunter ein Button "Abstimmen" - mit Sicherheitsabfrage, die den Text enthält "Hiermit stimmen Sie der Aussage <AA> zu" bzw. "... lehnen Sie die Aussage <AA> ab." bzw. "Hiermit enthalten Sie sich bezüglich der Aussage <AA>" bzw. "Hiermit wünschen Sie zu der Aussage <AA> eine Sitzung und beenden damit das Umlaufbeschlussverfahren" (o.ä. Texte).

Vor Erreichen dieser Seite muss sich das Mitglied authentifizieren (wie Sie das machen, überlasse ich Ihnen). Falls die Abstimmung vorgenommen wird (Ich stimme zu oder Ich stimme nicht zu) wird die Stimme notiert (namentlich) und die Radiobutton-Einheit + Button wird ersetzt durch einen nicht änderbaren Text "Sie haben der Aussage <AA> zugestimmt" bzw. "..abgelehnt" bzw. "Sie haben eine Sitzung zu diesem Thema beeantragt."

Falls ein Mitglied eine Sitzung beantragt sollte bei allen, die später auf die Abstimmungsseite kommen, auch kein Radiobutton mehr erscheinen, sondern der Text "Das Mitglied <Name> hat zu diesem Thema eine Sitzung beantragt." Auf diese Weise stimmt keiner mehr ab - aber die Materialien sind noch zugreifbar.

Falls der Vorsitzende die Abstimmung abbricht wird analog auch auf allen Seiten ein entsprechender Satz angezeigt. Wenn die Abstimmung gelaufen ist, dann steht statt den Interaktionselementen für die Wahl das Ergebnis da, inhaltlich wie in der o.a. Mail an die Mitglieder.

Bearbeiten der Gruppenmitglieder

Der Vorsitzende kann Gruppenmitglieder löschen und hinzufügen oder deren Angaben ändern als da sind:

- Name, Vorname, Titel
- Gruppenzugehörigkeit (freier kurzer Text, sowas wie Professor, Student)
- eMail-Adresse
- Haustelefon-Nummer
- Büroadresse (F2.122 o.ä.)

2. Pflichtenheft

Aus dem Lastenheft wurde in der ersten Phase des Projektes ein Pflichtenheft erstellt. Das Pflichtenheft ist das Ergebnis der Anforderungsanalyse. Gemeinsam mit den (genehmigten) Änderungsanträgen bilden die Anforderungen die Grundlage für das Visionsdokument.

Die Anforderungen werden tabellarisch in feingranularen Einzelregeln gepflegt, dies ist die sogenannte nicht-geschlossene Darstellung. Sie sind nach Komponenten und Prioritäten sortiert. Die Kompontenten sind A für *Admin* und V für *Vote*. Die Prioriäten heißen *1 (MUST)*, *2 (CAN)*, *3 (NICE)*. Einige Anforderungen betreffen mehrere Komponenten.

ID	BESCHREIBUNG	KOMP	PRIO
REQ011	P-T ist eine griffige Bezeichnung, erfasst in einer Zeile (HTML Input-Tag einzeilig).	A	1–MUST
REQ030	**Im System existiert eine Benutzerverwaltung samt Gruppendefinition. Eine Gruppe ist eine geschlossene Anzahl von Benutzern, die zusammengefaßt zu Wahlen eingeladen werden können. Zu einem Benutzer können Vorname, Familienname, ein Titel, eine Emailadresse, eine Telefonnummer und eine Büroadresse geführt werden. Weiterhin eine kurze Beschreibung (Professor, Student, Familie, Sport, Privat, Geschäftlich, etc).**	A	1–MUST
REQ015	Der Standardwert P-V sind zwei Werktage. Es wird eine Option angeboten eine Anzahl Werktage automatisch berechnen zu lassen. Die Angaben werden vom System validiert.	A	1–MUST
REQ012	P-D ist mehrzeiliger formatierter Text, Auszeichnungssprache ist BBCode.	A	1–MUST
REQ013	P-A können beliebige Dateien sein. Es gibt eine Obergrenze für die Dateigröße.	A	1–MUST
REQ016	Samstag ist im Sinne der Anwendung kein Werktag.	A	1–MUST
REQ023	Die automatisierte Verteilung des Ergebnisses kann im Falle der gezielten Verteilung durch den Administrator einen weiteren Kommentar enthalten, beispielsweise mit einer Quintessenz.	A	1–MUST

REQ021	Alle Wahlen (laufende, inaktive, abgebrochene und abgeschlossene) können in der Komponente A (Admin) geklont werden, um daraus neue Wahlen zu generieren.	A	1–MUST
REQ017	**Die Zugangszeit ist zusammenhängend.**	A	1–MUST
REQ018	Die Parameter einer Wahl können vom Administrator editiert werden.	A	1–MUST
REQ006	Der Workflow zur Initiierung einer Wahl besteht aus der (1) Definition der Wahl durch den Administrator, der (2) Veröffentlichung per sofort oder per Knopfdruck, dem (3) Versand einer Email an die Wähler mit dem Hinweis auf die Abstimmung und einem Zugangslink.	A	1–MUST
REQ005	Option zur zeitlichen Begrenzung einer Abstimmung und Sicherstellung, dass nur innerhalb des definierten Zeitraums abgestimmt werden kann.	A	1–MUST
REQ004	Auszeichnung des Administrators einer Wahl, er ist der Nutzer der Schnittstelle A. Der Administrator ruft zu einem Zeitpunkt zur Abstimmung auf.	A	1–MUST
REQ003	Option zur Definition der Wähler einer Wahl mit namentlicher Kennung und bekannter Emailadresse. Die Wähler sind die Nutzer der Komponente V. Es muss sicherstellbar sein, dass nur genau diese Wähler tatsächlich abstimmen. Der Vorsitzende kann auch Wähler sein, ist es aber nicht automatisch.	A	1–MUST
REQ010	Die Parameter einer Wahl sind ein Titel (P-T, Parameter Title), eine Beschreibung (P-D, Parameter Description), ein Kommentar (P-C, Patameter Comment), Attachments (P-A, Parameter Attachements), der Antragstext (P-R, Parameter Request), die Zugangszeit (P-V, Parameter Valid Time).	A	1–MUST
REQ008	Während der Zugangszeit einer Wahl können ihre Parameter mit Ausnahme des Kommentars nicht verändert werden.	A	1–MUST
REQ007	**Die Zugangszeit zu einer Wahl ist der Zeitraum zwischen ihrer Veröffentlichung (Aktivierung) und dem ersten Auftreten der vier Ereignisse (1) Erreichen des Endzeitpunktes, (2) Abbruch durch einen Teilnehmer, (3) Abbruch durch den**	A	1–MUST

	Administrator, (4) Abgabe aller möglichen Stimmen. Die Zugangszeit kann damit bei offenem Ende nicht (definiert) endlich sein.		
REQ031	Eine Wahl kann unterbrochen werden. Der definierte Endzeitpunkt verlängert sich um die Dauer des Unterbruchs.	A	2-CAN
REQ032	Der Administrator einer Wahl kann die Option erhalten, eine Abstimmung zu einem festen Termin starten zu lassen. Bei Erreichen dieser Zeit werden die üblichen Benachrichtungen versendet und die Wahl aktiviert.	A	3-NICE
REQ033	Bei der Beendigung einer Abstimmung kann das System optional eine Email an die Teilnehmer versenden. Dies ist eine Option in den Einstellungen jedes Benutzers.	A	3-NICE
REQ009	Nach Ende der Abstimmung wird unter dem zugehörigen Link ein Report angezeigt. Der Administrator erhält die Option diesen Report per Email automatisiert zu verteilen.	A, V	1–MUST
REQ002	Umsetzung als Zweikomponentensystem, einer Komponente zur Definition und Administration (Komponente A wie Admin) einer Wahl und eine zweite Komponente zur Erfassung der Stimmen pro Wahl (Komponente V wie Vote).	A, V	1–MUST
REQ001	Realisierung eines E-Voting-Systems zur Durchführung von Internetwahlen oder Web-Polls im Sinne von [W02].	A, V	1–MUST
REQ020	**Eine veröffentlichte Wahl kann während der Zugangszeit abgebrochen werden. Das System bietet dem Auslöser des Abbruches Begründungen an, wie die Forderung nach einer Sitzung zu dem Thema oder die nicht mehr gegebene Sinnhaftigkeit der Wahl. Der Administrator kann jede Wahl abbrechen. Das System bietet die Option den Abbruch jedem Teilnehmer zu erlauben (P-B, Parameter Break).**	A, V	1–MUST
REQ019	Die Email zur Veröffentlichung einer Wahl geht an die Wähler und den Administrator. Sie enthält den Link zur Wahl, sowie P-V, P-T, P-R, dabei steht P-T im Betreff, Reply-To ist der Administrator.	A, V	1–MUST

REQ014	Über P-R wird in der Wahl abgestimmt. Der Text wird nicht internationalisiert, sondern in der exakten Form wiedergegeben, in der er vom Administrator definiert wurde.	V	1–MUST
REQ025	Das Absenden der Abstimmung kann vom System mit den gewählten Optionen hinterfragt werden (P-S, Parameter Sure).	V	1–MUST
REQ022	**Ein Report enthält die Angabe über die tatsächlichen Teilnehmer an der Wahl (Anzahl und Namen), das Ergebnis oder ein Zwischenergebnis und die Information über den Abbruch.**	V	1–MUST
REQ024	Die Kompontente V (Vote) zeigt den Wählern P-T, P-D, P-C als Texte und die P-A als Download-Links, eine Einheit Radio-Buttons (ohne Voreinstellung) mit den Abstimmungsalternativen und optional einem Button zum Abbrechen der Wahl.	V	1–MUST
REQ026	Ein Wähler kann sich authentifizieren oder gegebenfalls sogar autorisieren müssen (P-AU, Parameter Autherisation).	V	1–MUST
REQ027	Die Oberfläche zeigt nach Abgabe der Stimme die gewählte Option an und ersetzt damit die Radio-Button-Einheit.	V	1–MUST
REQ028	Im Falle des Abbruches bleibt die Wahl für alle Wähler sichtbar, die Attachments bleiben zugreifbar. Es wird über den Status der Wahl informiert.	V	1–MUST
REQ029	Nach Ablauf der Wahl zeigt die Wahlseite den Report der Wahl.	V	1–MUST

3. Change-Requests

In diesem Anhang werden die Change-Requests aufgelistet, wie sie in Ursache und Wirkung im Kapitel Change-Management über den Lebenszyklus des Rational Unified Process beschrieben wurde. Jeder Change-Request betrifft genau eine Anforderung.

CR01: Keine Cron-Jobs

Anforderung: REQ006

Alte Formulierung: Der Workflow zur Initiierung einer Wahl besteht aus der (1) Definition der Wahl durch den Administrator, der (2) Veröffentlichung per voreingestellter Zeit oder per Knopfdruck, dem (3) Versand einer Email an die Wähler mit dem Hinweis auf die Abstimmung und einem Zugangslink.

Neue Formulierung: Der Workflow zur Initiierung einer Wahl besteht aus der (1) Definition der Wahl durch den Administrator, der (2) Veröffentlichung per sofort oder per Knopfdruck, dem (3) Versand einer Email an die Wähler mit dem Hinweis auf die Abstimmung und einem Zugangslink.

Begründung: Keine Cron-Jobs vorgesehen

Status: Genehmigt

CR02: Definition Zugangszeit

Anforderung: REQ007

Alte Formulierung: Die Zugangszeit zu einer Wahl ist der Zeitraum zwischen der Veröffentlichung einer Wahl und dem ersten Auftreten eines Abbruches oder dem definierten Ende der Wahl. Die Zugangszeit kann damit bei offenem Ende nicht (definiert) endlich sein.

Neue Formulierung: Die Zugangszeit zu einer Wahl ist der Zeitraum zwischen ihrer Veröffentlichung (Aktivierung) und dem ersten Auftreten der vier Ereignisse (1) Erreichen des Endzeitpunktes, (2) Abbruch durch einen Teilnehmer, (3) Abbruch durch den Administrator, (4) Abgabe aller möglichen Stimmen. Die Zugangszeit kann damit bei offenem Ende nicht (definiert) endlich sein.

Begründung: Klarere Formulierung unter Berücksichtigung aller Beendigungsmöglichkeiten.

Status: Beantragt

CR03: Keine endegesteuerten Ereignisse

Anforderung:	REQ009
Alte Formulierung:	Nach Ende der Wahl (entweder durch Abbruch oder Ablaufs der Zugangszeit) wird aus dem Abstimmungsergebnis ein Report generiert, über die Generierung wird der Administrator per Email informiert. Der Report kann vom Administrator automatisiert an die Wähler verteilt werden.
Neue Formulierung:	Nach Ende der Abstimmung wird unter dem zugehörigen Link ein Report angezeigt. Der Administrator erhält die Option diesen Report per Email automatisiert zu verteilen.
Begründung:	Keine Cron-Jobs, Nomenklatur Abstimmung statt Wahl
Status:	Beantragt

4. Testfälle

Der Anhang listet in loser Reihenfolge die Testfallbeschreibungen auf. Jede Testfallbeschreibung besteht aus einem Titel und den vier Absätzen Testdaten, Vorbedingungen, Testschritte, Nachbedingungen.

TF01: Generelle Funktionalität

Testdaten

Titel: Erster Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Web-Poll
Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: Keine
Weitere Teilnehmer: Keine
Anforderungen: REQ001, REQ002, REQ004, REQ025
Methode: Smoke-Test

Vorbedingungen

Im System muss ein Mitglied registriert, verifiziert und angemeldet sein.

Testschritte

Die Seite create.php aufrufen und im dort erscheinenden Dialog die Testdaten einfügen. Durch Auslösen des Buttons „Senden“ die Eingabe bestätigen. Das System zeigt eine Zusammenfassung der eingegeben Werte an. Die Nachfrage bestätigen. Die Abstimmung aufrufen. Dann über den Workflow „Abstimmen“ den Web-Poll mit einer „Ja“-Stimme füttern. Die Abstimmung wieder aufrufen (Seite view.php).

Nachbedingungen

Auf der Seite der Abstimmung werden die Daten wie eingegeben dargestellt. Start- und Endzeiten sind korrekt. Die Abstimmung wird mit mit einer abgegebenen Stimme korrekt geführt.

TF02: Teilnehmerauswahl

Testdaten

Titel: Zweiter Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Umfrage

Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Peter Lustig (peter.lustig@pbci.de)
Uli Hoeneß (wuerstchen-uli@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Anforderungen: REQ003, REQ004
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Die unter Testdaten aufgeführten Mitglieder und zusätzlich der Benutzer Nikolaus Risch (nikolaus.risch@pbci.de) sind im System als Benutzer registriert und verifiziert sein. Das Mitglied John Wayne ist angemeldet. Die Mitglieder Peter Lustig und Uli Hoeneß sind im Adressbuch von John Wayne.

Testschritte
Die Seite create.php aufrufen und im dort erscheinenden Dialog die Testdaten einfügen. Anschließend unter view.php die Abstimmung betrachten und abstimmen. John Wayne abmelden und Peter Lustig anmelden. Abstimmen und abmelden. Schließlich Uli Hoeneß anmelden und erneut abstimmen. Zuletzt auf Joseph Ratzingen ummelden und zum vierte Mal eine Stimme abgeben. Zuletzt noch auf Nikolaus Risch ummelden und die Seite betrachten.

Nachbedingungen
Alle Abstimmungen waren möglich und werden in der Abstimmung angezeigt. Der Benutzer Nikolaus Risch bekommt den Workflow „Abstimmen“ nicht angezeigt, sondern einen Hinweis, dass Abstimmen für ihn nicht möglich ist.

TF03: Aktivierte Abstimmungen

Testdaten
Titel: Dritter Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Wahl
Dauer: 9 Werktage über ein Wochenende
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: Peter Lustig (peter.lustig@pbci.de)
Uli Hoeneß (wuerstchen-uli@pbci.de)
Weitere Teilnehmer: Keine
Anforderungen: REQ005, REQ006, REQ008, REQ015, REQ016, REQ017
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Die Benutzer John Wayne, Peter Lustig, Uli Hoeneß sind am System registriert und verifiziert. John Wayne ist angemeldet.

Testschritte
Eine Abstimmung anlegen, aber noch nicht aktivieren. Auf die Seite gehen und verifizieren, dass eine Abstimmung nicht möglich ist. Es wird der Status „Inaktiv“ angezeigt. Die Abstimmung aktivieren, abstimmen und abbrechen. Ausloggen und als Benutzer Peter Lustig anmelden. Die Abstimmung besuchen und verifizieren, dass ein Abstimmen nicht möglich ist.

Nachbedingungen
Keine.

TF04: Abstimmungsende

Testdaten

Titel:	Vierter Funktionstest [Untertestnummer]
Fragestellung:	Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung:	Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung:	Umfrage
Dauer:	1 Tag
Benutzerabbruch:	Erlaubt
Eingeladene Mitglieder:	John Wayne (john.wayne@pbci.de)
Weitere Teilnehmer:	joseph.ratzinger@pbci.de
Anforderungen:	REQ007
Methode:	Anforderungsbasiertes Testen

Vorbedingungen
John Wayne ist am System angemeldet.

Testschritte
Vier Abstimmungen nach dem Muster der Testdaten anlegen, den Titel zur Eindeutigkeit mit einer Nummer kennzeichnen. Abstimmung 01 abbrechen. Ausloggen und als Wayne und Ratzinger einloggen und für Abstimmung 02 abstimmen. Abstimmung 03 abbrechen. Systemzeit auf dem Server einen Tag vorstellen.

Nachbedingungen
Alle vier Abstimmungen befinden sich im Zustand beendet und zwar in der Reihenfolge Abgebrochen (Administrator), Abgestimmt, Abgebrochen (Teilnehmer), und Ausgelaufen.

TF05: Informationsemailversand

Testdaten

Titel:	Fünfter Funktionstest
Fragestellung:	Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung:	Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.

Zugangsbeschränkung: Umfrage
Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Anforderungen: REQ006, REQ009, REQ019, REQ023
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Der Benutzer John Wayne ist am System angemeldet.

Testschritte
Eine Abstimmung gemäß Vorgabe anlegen und aktivieren. Beide Teilnehmer abstimmen lassen. Als John Wayne einloggen und Report versenden.

Nachbedingungen
Die Emailadresse joseph.ratzinger@pbci.de erhält zwei Emails, die erste mit der Einladung und die zweite mit dem Report. John Wayne erhält als Administrator keine Emails.

TF06: Abstimmung bearbeiten

Testdaten
Titel: Sechster Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Umfrage
Dauer: 2 Tage
Start: Manuell
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: Peter Lustig (peter.lustig@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Administrator: John Wayne (john.wayne@pbci.de)
Anforderungen: REQ008, REQ010, REQ018
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Das Mitglied John Wayne ist angemeldet, Peter Lustig ist registriert und verifiziert.

Testschritte
Eine Abstimmung mit den Testdaten erstellen. Anschließend unter Bearbeiten Titel, Fragestellung und Beschreibung verändern, Benutzerabbruch verbieten und den Typ von Umfrage auf Wahl umstellen. Bestätigen.

Nachbedingungen
Die Übersicht über die Abstimmung zeigt die veränderten Daten korrekt an.

TF07: Report

Testdaten

Titel: Zweiter Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Umfrage
Dauer: 1 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Peter Lustig (peter.lustig@pbci.de)
Uli Hoeneß (wuerstchen-uli@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Administrator: John Wayne (john.wayne@pbci.de)
Anforderungen: REQ009, REQ022, REQ023, REQ029
Methode: Anforderungsbasiertes Testen

Vorbedingungen

Die Mitglieder Wayne, Lustig und Hoeneß sind am System registriert und verifiziert.

Testschritte

Die Abstimmung wie beschrieben anlegen. Die Teilnehmer abstimmen lassen. Die Systemzeit auf dem Server um einen Tag vorstellen. Die Seite besuchen und einen Report samt Kommentar versenden.

Nachbedingungen

An die Emailadressen von Lustig, Hoeneß und Ratzinger wird eine Email versendet mit dem Link auf den Report und den Angaben über die tatsächlichen Teilnehmer namentlich, das Ergebnis samt Anzahl abgegebener Stimmen und den Kommentar. Der Kommentar wird weiterhin auf der Seite des Reports vermerkt.

TF08: Grenzwertanalyse

Testdaten

Titel: 1 bis 60 Zeichen
Fragestellung: 1 bis 320 Zeichen
Beschreibung: 0 bis 1200 Zeichen
Zugangsbeschränkung: Web-Poll
Dauer: Ganze Zahl von 1 bis 3650
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: Keine
Weitere Teilnehmer: Keine
Anforderungen: REQ010, REQ011, REQ012, REQ013, REQ014, REQ015
Methode: Anforderungsbasiertes Testen

Vorbedingungen

Keine

Testschritte
Der Testfall überprüft die Eingabebedingungen von Titel, Fragestellung, Beschreibung und Dauer. Dazu bei jedem der genannten Felder die Kombinationen an den Grenzen und ober- beziehungsweise unterhalb testen (Grenzwertanalyse). Im Beispiel des Titels also einen leeren Titel (Fehler) einen Titel der Länge 1, der Länge 60 und der Länge 61 (Fehler). Bei der Dauer zusätzlich einen Text eingeben, der keine Zahl darstellt.

Nachbedingungen
Die Eingabevalidierung soll im Fehlerfall den notwendigen Umweg der Neueingabe korrekt präsentieren.

TF09: Abstimmung abbrechen

Testdaten

Titel: Neunter Funktionstest 01/02
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Wahl
Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Weitere Teilnehmer: Keine
Administrator: Peter Lustig (peter.lustig@pbci.de)
Anforderungen: REQ020, REQ028
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Die Mitglieder Wayne und Lustig sind am System registriert und verifiziert.

Testschritte
Die Abstimmung mit den genannten Testdaten zweimal anlegen und in einem Fall vom Wayne und im Zweiten Fall von Lustig abbrechen lassen. Begründungen eingeben.

Nachbedingungen
Auf den Seiten der Abstimmungen werden der jeweilige Abbruch und die Begründungen für die Abbrüche angezeigt. Die Daten der Wahl inklusive der Anhänge bleiben sichtbar.

TF10: Abstimmung klonen

Testdaten

Titel: Zehnter Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?

Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Umfrage
Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Peter Lustig (peter.lustig@pbci.de)
Uli Hoeneß (wuerstchen-uli@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Anforderungen: REQ021
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Die Mitglieder sind am System registriert und verifiziert.

Testschritte
Die Abstimmung wie beschrieben erstellen und anschließend mit einem neuen Titel klonen.

Nachbedingungen
Die geklonte Abstimmung wird korrekt angezeigt.

TF11: Zugangsbeschränkung

Testdaten
Titel: Elfter Funktionstest
Fragestellung: Finden Sie das System so benutzbar und wirklich wunderschön?
Beschreibung: Die gewählte Frage dient nur der Überprüfung der Grundfunktion und ist kein Zeichen von drohender Hybris des Autors.
Zugangsbeschränkung: Variabel
Dauer: 2 Tage
Benutzerabbruch: Erlaubt
Eingeladene Mitglieder: John Wayne (john.wayne@pbci.de)
Peter Lustig (peter.lustig@pbci.de)
Weitere Teilnehmer: joseph.ratzinger@pbci.de
Administrator: John Wayne (john.wayne@pbci.de)
Anforderungen: REQ026 + mehr!
Methode: Anforderungsbasiertes Testen

Vorbedingungen
Die Mitglieder Wayne, Lustig und Hoeneß sind am System registriert und verifiziert.

Testschritte
Eine Abstimmung mit den Testdaten als Wahl anlegen. Dort auf den weiteren Teilnehmer verzichten. Als Joseph Ratzinger anmelden und die Seite der Abstimmung besuchen. Die Abgabe der Stimme ist nicht möglich. Die Abstimmung noch einmal als Umfrage anlegen. Als Gast die Seite besuchen und

mit dem Code von Ratzinger abstimmen. Die Abstimmung als Meinungsbild anlegen und als Gast abstimmen.

Nachbedingungen
Die Stimmen wurden korrekt erfasst.

TF12: Adressbuch

Testdaten

Titel:	Kein
Fragestellung:	Keine
Beschreibung:	Keine
Zugangsbeschränkung:	n/a
Dauer:	n/a
Benutzerabbruch:	n/a
Eingeladene Mitglieder:	Keine
Weitere Teilnehmer:	Keine
Administrator:	John Wayne (john.wayne@pbci.de)
Anforderungen:	REQ030
Methode:	Anforderungsbasiertes Testen

Vorbedingungen
Das Mitglied John Wayne ist am System registrtriert, verifiziert und angemeldet. Die Mitglieder Peter Lustig (peter.lustig@pbci), Uli Hoeneß (wuerstchen-uli@pbci.de) und Joseph Ratzinger (joseph.ratzinger@pbci.de) sind am System registriert und verifiziert.

Testschritte
Die Mitglieder Ratzinger und Hoeneß zur Gruppe Topmanager hinzufügen. Als John Wayne die Gruppe Topmanager und das Mitglied Lustig zum Adressbuch hinzufügen.

Nachbedingungen
Bei der Anlage einer neuen Abstimmen erscheint die Gruppe und alle drei Mitglieder im Adressbuch.

5. Literaturverzeichnis

Die Quellen für eine Arbeit im Umfeld der Informationstechnologie liegen naturgemäß gedruckt *und* digital vor. Aufgrund der Aktualität der Themen lassen sich darüber hinaus wichtige Beiträge in Fachartikeln diverser Zeitschriften finden.

Neben klassischen Quellen in Form gedruckter Bücher macht diese Arbeit daher Gebrauch von Quellen in Fachmagazinen und von Internetquellen. Letztgenannte Dokumente unterscheiden sich dadurch, dass sich ihre Verfügbarkeit ändern kann, indem ihre Autoren sie löschen oder sich die Zugangsadressen ändern. Eine besondere Plattform für Wissen stellt das Online-Lexikon Wikipedia dar. Durch die Beteiligung vieler Autoren an einem Artikel entsteht eine objektive Sicht.

Es gibt Diskussionen über die Qualität und Zitierfähigkeit von Internetquellen und Wikipedia-Artikeln im Vergleich zu den klassischen Quellen und Enzyklopädien und über die Art und Weise, wie diese Quellen zitiert und im Literaturverzeichnis aufgeführt werden können [O1], [W1], [P4]. Als Quintessenz unterscheidet diese Arbeit zwischen gedruckten Werken, Fachartikeln, online verfügbaren Beiträgen und Artikeln der Wikipedia. Fachlich relevante Elemente findet sich überwiegend in gedruckten Werken, allerdings ist die Qualität der Onlinequellen, insbesondere in Wikipedia und hier insbesondere bei Stichpunkten zur Webentwicklung mitunter so gut, dass die Arbeit auf dies Hinweise nicht verzichten kann. Quellen anderer Internetdienste wie Newsgroups werden nicht verwendet.

Die Reihenfolge der Auflistung ist zufällig, sie stellt keine Wertung dar und folgt nicht der Chronologie des ersten Zitierens. Alle Internationalen Standard Buch Nummern wurden in nach ISBN-13 konvertiert, auch wenn die Originalausgabe noch vor 2006 und mit ISBN-10 erfolgte.

5.1 Gedruckte Werke

[P01] **Dudenredaktion (Herausgeber),** *Deutsches Universalwörterbuch*, 6. Auflage, Bibliographisches Institut 2006, Mannheim, ISBN 978-3-411-05506-7.

[P02] **Philippe Kruchten,** *The Rational Unified Process An Introduction*, Second Edition, Addison Wesley 2000, ISBN 978-0-201-70710-6.

[P03] **Phillippe Kruchten**, *Der Rational Unified Process – Eine Einführung*, Übersetzung von Cornelia Versteegen, Addison-Wesley München 1999, ISBN 978-3-8273-1543-4. Deutsche Übersetzung der ersten Auflage. Einige direkte Übersetzungen verändern den Sinn der Aussage. Daher werden beide Versionen verwendet.

[P04] **Jens Runkehl und Thorsten Siever**, *Das Zitat im Internet - Ein Electronic Style Guide zum Publizieren, Bibliografieren und Zitieren*, 3. Auflage, Revonnah Verlag, Hannover 2000,

ISBN 978-3-927715-83-7.

[P05] **Jan Skrobotz**, *Das elektronische Verwaltungsverfahren*, Dissertation, Duncker & Humblot, Berlin 2005, ISBN 978-3-428-11647-8.

[P06] **Tobias Gantert**, *E-Demokratie – Chancen und Risiken*, CDM Verlag Dr. Müller, Saarbrücken 2006, ISBN 978-3-86550-793-8.

[P07] **Booz Allen Hamilton (Herausgeber)**, *E-Government und der moderne Staat*, F.A.Z.-Institut für Management-, Markt- und Medieninformationen, Frankfurt 2002, ISBN 978-3-934191-50-1.

[P08] **Prof. Dr. Johannes Siedersleben**, *Moderne Software-Architektur: Umsichtig planen, robust bauen mit Quasar,* Dpunkt Verlag, Heidelberg 2004, ISBN 978-3-89864-292-7.

[P09] **Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides**, *Entwurfsmuster – Elemente wiederverwendbarer objektorientierter Software*, Addison-Wesley, München 1996, ISBN 978-3-8273-1862-6.

[P10] **Eric Freeman, Elisabeth Freeman**, *Head First Design Patterns*, O'Reilly, Sebastopol 2004, ISBN 978-0-596-00712-6.

[P11] **Dr. Wolfgang Büchner, Prof. Dr. Alfred Büllesbach (Herausgeber)**, *E-Government*, Deutsche Gesellschaft für Recht und Informatik e.V. im Verlag Dr. Otto Schmidt, Köln 2003, ISBN 978-3-504-67011-5.

[P12] **Martin Will**, *Internetwahlen – verfassungsrechtliche Möglichkeiten und Grenzen*, Reihe Recht und neue Medien, Boorberg Verlag, Stuttgart 2002, ISBN 978-3-415-03082-4.

[P13] **Prof. Dr. Bernd Holznagel, Dr. Andreas Grünwald, Anika Hanßmann**, *Elektronische Demokratie*, Schriftenreihe Demokratie und Recht, Verlag C.H. Beck, München 2001, ISBN 978-3-406-48155-0.

[P14] **Robert Schönau**, *Elektronische Demokratie – verfassungsrechtliche Zulässigkeit elektronischer Wahlen*, Reihe Verfassungsrecht in Forschung und Praxis, Band 37, Kovač Verlag, Hamburg 2007, ISBN 978-3-8300-2666-2.

[P15] **Klaus Koziol, Gerfried W. Hunold (Herausgeber)**, *EDemokratie = Ende der Demokratie?*, Schriftenreihe Forum Medienethik der Diözese Rottenburg-Stuttgart und der Universität Tübingen, Kopaed Verlag, München 2001, ISBN 978-3-935686-07-5.

[P16] **Dirk W. Hoffmann**, *Software-Qualität*, Springer-Verlag, Berlin und Heidelberg 2008, ISBN 978-3-540-76322-2.

[P17] **Per Kroll, Bruce MacIsaac**, *Agility and Discipline Made Easy – Practices from OpenUP and RUP,* Addison-Wesley, München 2006, ISBN 978-0-321-32130-5.

[P18] **David Flanagan**, *JavaScript – Das umfassende Referenzwerk*, 4. Auflage, Deutsche Ausgabe, O'Reilly, Köln 2002, ISBN 978-3-89721-330-2.

[P19] **Prof. Dr. Ulrich Karpen**, *Elektronische Wahlen? - Einige verfassungsrechtliche Fragen*, Nomos Verlagsgesellschaft, Baden-Baden 2005, ISBN 978-3-8329-1249-9.

[P20] **Esfandiar Khorrami**, *Bundestagswahlen per Internet – Zur rechtlichen und tatsächlichen Realisierbarkeit von Internetwahlen*, Nomos Verlagsgesellschaft, Baden-Baden 2006, ISBN 3-8329-1530-3.

[P21] **Andreas Spillner, Thilo Linz**, *Basiswissen Softwaretest*, 3. Auflage, Dpunkt Verlag, Heidelberg 2005, ISBN 3-89864-358-1.

[P22] **Christian Bunse, Antje von Knethen**, *Vorgehensmodelle kompakt*, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg 2008, ISBN 978-3-8274-1950-7.

[P23] **Andy Clarke, Molly Holzschlag**, *Transcending CSS – Neue kreative Spielräume im Webdesign*, Addison-Wesley Verlag, München 2007, ISBN 978-3-8273-2545-7.

[P24] **Heide Balzert**, *UML 2 Kompakt*, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg 2005, ISBN 3-8274-13893.

[P25] **Alexander Siedschlag, Alexander Bilgeri, Dorothea Lamatsch (Herausgeber)**, *Kursbuch Internet und Politik – Elektronische Demokratie und virtuelles Regieren*, Band 1/2001, Leske und Budrich Verlag, Opladen 2001, ISBN 3-8100-3309-X.

5.2 Magazine

[M01] **HMD – Praxis der Wirtschaftsinformatik**, M*ake or Buy in der IT*, Heft 206, April 1999, Dpunkt Verlag, Heidelberg, ISSN 1436-3011.

5.3 Beiträge der Wikipedia

[W01] **Zitieren von Internetquellen**. Beitrag über den Umgang mit Zitaten aus dem World-Wide-Web. Stand vom 30. Juni 2007.

[W02] **E-Voting**. Auflisting von verschiedenen Ausprägungen des E-Voting als Teildisziplin von E-Democracy. Stand vom 17. Februar 2008.

[W03] **Klickibunti**. Erklärung des sehr ungangssprachlichen Wortes und Legitimation für seine Verwendung. Stand vom 23. Februar 2008.

[W04] **E-Government**. Umfassenste Definition des Begriffes mit Einordnung von und Verweisen auf Unterbegriffe.

[W05] **E-Administration**. Betrachtung von E-Government im engeren

Sinne. Die Online-Dokumentation fasst Definitionen des Begriffes als Teilmenge von E-Government im weiteren Sinne zusammen.

[W06] **E-Democracy**. Betrachtung von E-Democracy als Teilmenge von E-Government und Überbegriff für E-Participation und E-Voting.

[W07] **Chaos-Studie**. Beschreibung des Chaos-Reports der Standish-Group. Quelle der Zahlenbasis ist der frei verfügbare Report aus dem Jahr 1994.

[W08] **Modellgetriebene Softwareentwicklung**. Zusammenfassung des Model-Driven-Development (MDD).

[W09] **Feature-Driven Development**. Darstellung einiger Aspekte des FDD.

[W10] **Rational Unified Process**. Kurzer Überblick über die Nomenklatur des RUP. Wesentlich ausführlicher ist die englische Version des Artikels.

[W11] **JavaScript**. Einführung samt Geschichte und Versionsgeschichte der vorherschenden clientseitigen Websprache.

[W12] **Abstimmung**. Schwammige Definition von Abstimmung als Stellungnahme zu Anträgen mit den Optionen Ja, Nein und Enthaltung.

[W13] **TED**. Artikel über Tele-Voting-Verfahren im Allgemeinen und den vom ZDF 1979 entwickelten Tele-Dialog im Besonderen.

5.4 Internetquellen

[O01] **Dittmar Graf**, *Das Internet – zum Zitieren geeignet*, www.uni-giessen.de, Pfad: /biodidaktik/vdbiol/zitieren.htm, zuletzt gesehen im Juni 2007.

[O02] **Viele Autoren**, *Wikipedia – Die freie Enzyklopädie*, de.wikipedia.org, zuletzt gesehen im September 2007.

[O03] **Stefan Münz,** *SELFHTML – Die Energie des Verstehens*, de.selfhtml.org, zuletzt gesehen im September 2007. Selfhtml ist wohl die wichtigste Referenz für Sprachen, die vom Browser interpretiert werden. Der Autor Stefan Münz lehnt sich eng an die Vorgaben des *World-Wide-Web-Consortiums* an und unterlegt die Elemente mit Beispielen.

[O05] **Studierendenverzeichnis,** http://www.studivz.net. Soziale Netzwerkanwendungen mit sicherheitskritischen Übertragung von Benutzerdaten im Klartext.

[O06] **Aktionsbündnis gegen Studiengebühren**, http://www.abs-bund.de. Protestseite vom ABS unter der Dachorganisation des Freien Zusammenschlusses der Studierendenschaften (FZS). Beispiel für EProtest. *Zuletzt gesehen im Februar 2009*.

[O07] **Studiengebuehren.de**, http://www.studiengebuhren.de. Aktion des

Ring Christlich-Demokratischer Studenten (RCDS) als Antwort auf das Aktionsbündnis gegen Studiengebühren. Beispiel von E-Discussion. *Zuletzt gesehen im Februar 2009*.

[O08] **Global Centre for ICT in Parliament**, http://www.ictparliament.org. Unterorganisation der Vereinten Nationen zur Untersuchung des Potentials von „Information and Communication Technologies (ICT)“ für parlamentarische Arbeit. *Zuletzt gesehen im März 2009*.

[O09] **World E-Parliament Conference 2008**, http://www.ictparliament.org/worldeparliamentconference2008/. Konferenz über E-Parliament. Im Zuge der Konferenzergebnisse Herausgabe des World E-Parliament Report. *Zuletzt gesehen im März 2009*.

[O10] **The Standish Group**, www.standishgroup.com, Herausgeber des seit 1994 erstmals erhobenen Chaos-Reports zur Untersuchung von Erfolgs- und Mißerfolgsfaktoren von Softwareprojekten. Mit regelmäßigen Aktualisierungen. *Zuletzt gesehen im März 2009*

[O11] **Feature-Driven Development**, featuredrivendevelopment.com. Homepage der Vertreter des FDD. Enthält Links, Foren, Buchempfehlungen und die Darstellung von Zertifizierungsmöglichkeiten. *Zuletzt gesehen im März 2009*.

[O12] **U.S. Food And Drug Administration**, www.fda.gov. Webseite der FDA. Unter „Medical Devises“ werden Zulassungsregulatorien für den Einsatz von Software in der medizinischen Diagnose angegeben. Demnach müssen insbesondere Softwarekomponenten von Drittanbietern eigenen Qualitätssicherungsmerkmalen unterzogen werden. *Zuletzt gesehen im März 2009*.

[O13] **Gerd Szwillus**, Usability Engineering, http://wwwcs.upb.de, Pfad: /cs/ag-szwillus/lehre/lehre.html. *Zuletzt gesehen im September 2006*.

[O14] **International Software Testing Qualifications Board (ISTQB)**, http://www.istqb.org, Webpräsenz der internationale Zertifizierungsstelle für Tester. *Zuletzt gesehen im März 2009*.

[O15] **Serena**, http://www.serena.com, Webpräsenz der Firma Serena, einem Hersteller von Softwareprojektmanagementwerkzeugen. Serena entwickelt das Konzept der Business Mashups. *Zuletzt gesehen im März 2009*.

[O16] **Polarion**, http://www.polarion.com. Webpräsenz der Firma Polarion, einem Anbieter eines integrierten Systems zur Steuerung eines Softwareprojektes. *Zuletzt gesehen im März 2009.*

[O17] **PHPEclipse**, http://www.phpeclipse.com. Seite der auf Eclipse basierenden Entwicklungsumgebung für PHP. *Zuletzt gesehen im Oktober 2008.*

[O18] **PHP Development Tools Project**, http://www.eclipse.org/pdt/.

Dokumentation und Binaries von PDT, der von Zend und IBM unterstützten IDE für PHP. *Zuletzt gesehen im März 2009.*

[O19] **Webserver On Stock**, http://www.chsoftware.net/. Eine Software, welche einen Apache-Webserver als portable Software starten kann. Das Projekt wurde mittlerweile unter dem Namen Movable-Web-Server (MoWeS) als Open-Source freigegeben. *Zuletzt gesehe n im März 2009*.

[O20] **7-Zip**, http://www.7-zip.org. Freier Packer mit einer portablen Version. *Zuletzt gesehen im März 2009*.

[O21] **Doxygen**, http://www.doxygen.org. Sprachunabhängiges Werkzeug zur Codedokumentation nach dem Vorbild von JavaDoc. *Zuletzt gesehen im März 2009*.

[O22] **CSS Zen Garden**, http://www.csszengarden.com. Portal für Design und Webdesign mittels Cascading-Style-Sheets. *Zuletzt gesehen im März 2009*.

[O23] **Bundesverfassungsgericht**, http://www.bverfg.de, Pfad: /entscheidungen/cs20090303_2bvc000307.html. Urteil über den Einsatz von Wahlcomputern. Zitierung: BVerfG, 2 BvC 3/07 vom 3.3.2009, Absatz-Nr. (1-163). *Zuletzt gesehen im März 2009*.

[O24] **Forschungsgruppe Internetwahlen**, http://internetwahlen.de. Von der Bundesregierung eingesetzte Forschungsgruppe; Ersteller der Wahlsoftware i-vote. *Zuletzt gesehen im März 2009*.

[O25] **SafeVote**, http://www.safevote.com. *Zuletzt gesehen im März 2009*.

[O26] **VoteHere**, http://www.votehere.net. *Zuletzt gesehen im März 2009.*

[O27] **Fragebogen-Tool**, http://www.fragebogen-tool.de. Kostenlose Möglichkeit Umfragen zu erstellen. *Zuletzt gesehen im März 2009*.

[O28] **Vote24**, http://www.vote24.de. Angebot zur Erstellung von Web-Polls für die eigene Homepage. *Zuletzt gesehen im März 2009*.

[O29] **Polyas**, http://www.polyas.com. Anbieter von Wahlsoftware, unter anderem einem Abstimmungssystem. *Zuletzt gesehen im März 2009*.

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen. Diese Arbeit hat in dieser oder einer ähnlichen Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Paderborn, den 30. März 2009

Michael Köster